

A weakly universal cellular automaton on the pentagrid with two states

Maurice MARGENSTERN

February 16, 2016

Abstract

In this paper, we prove that there is a weakly universal cellular automaton on the pentagrid with two states. This paper improves in some sense a previous result with three states. Both results make use of *à la Moore* neighbourhood. However, the result with three states is rotation invariant while that with two states is not. In both cases, at each step of the computation, the set of non quiescent states has always infinitely many cycles.

1 Introduction

Self-contentedness is a convenience for the reader: he/she is exempted from searching definitions, statements and sometimes proofs from previous papers, many times from the same author(s). Now, if I comply to this constraint, anti-plagiarism software will warn the reader that there is a definite percent of overlap of this paper with other papers of the author. The struggle against plagiarism is definitely necessary. I just mention that it may have some inconvenience. Now, I am convinced that the reader will understand these inconveniences, so that he/she will play the game of looking into the papers indicated in the references instead of throwing the paper because some context is not at hand. Accordingly, I do not repeat most of the introduction of [9]. As mentioned in the abstract, the improvement with respect to [9] is obtained by relaxing the constraint of rotation invariance. Again, we repeat that the new result is a true planar cellular automaton, which means that at each step of the computation, the set of non quiescent states has always infinitely many cycles, as already mentioned in the abstract. Also, the reason for not repeating the introduction is that we make use of the same model of railway computation. Also for this reason, we do not mention the reminder sections of [9] about the railway model and its implementation in the hyperbolic tiling we use in this paper. We also not repeat the introduction to the approach to hyperbolic tilings used in this paper, again, the reader is referred to [2, 3, 5] and to several papers as [10] where other references are available.

In Section 2, I indicate the new features for implementing the model with the constraint of using two states only. In Section 3, we give the rules of the automaton, insisting in the way we defined these rules in a context where rotation invariance is no more required, which gives a definite advantage as indicated by the result itself:

Theorem 1 *There is a weakly universal cellular automaton on the pentagrid with Moore neighbourhood which is truly planar and which has two states.*

Presently, we turn to the proof of this result.

2 The scenario of the simulation

As in previous papers, sensors and control devices are no more immediate neighbours of the cells where tracks arrive at the point which define a switch. This reinforces the careful study of the tracks themselves as their role for conveying key information is more and more important.

Here too, tracks are blank cells marked by appropriate milestones. We carefully study this point in Sub-section 2.1. Later, in Sub-sections 2.2 and 2.3, we look at the changes introduced with respect to [9].

2.1 The tracks

In this implementation, the tracks are represented in the same way as in [9]. For the convenience of the reader, we repeat the illustration of this implementation given by Figure 1.

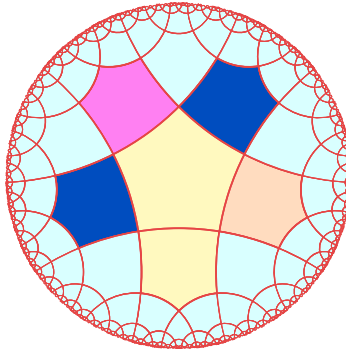


Figure 1 *Element of the tracks.*

However, some care has to be given to this structure. With two states, we have to thoroughly check that all constraints we have to define horizontal and vertical tracks can be satisfied. We refer the reader to Sub-section 3.1 where we give the rules and in which appropriate figures allow us to check the correctness of the implementation given in this paper. We just remember that for this

implementation, the locomotive is implemented as a single black cell: it has the same colour as the milestones of the tracks. Only the position of the locomotive with respect to the milestones allow us to distinguish it from the milestones. As clear from the next sub-section, we know that besides this **simple locomotive**, the locomotive also occurs as a **double one** in some portions of the circuit. Moreover, the circuit also makes use of signals which are implemented in the form of a simple locomotive. So that at some point, it may happen that we have three simple locomotives travelling on the circuit: the locomotive and two auxiliary signals involved in the working of some switch. For aesthetic reasons, the black colour which is opposed to the blank is dark blue in the figures.

2.2 The round-about

The round-about replaces the crossing, a railway structure, by a structure inspired by road traffic. At a round-bout where two roads are crossing, if you want to keep the direction arriving at the round-about, you need to leave the round-about at the second road. Figure 2 illustrates this features, a figure from [9] which is repeated here for convenience.

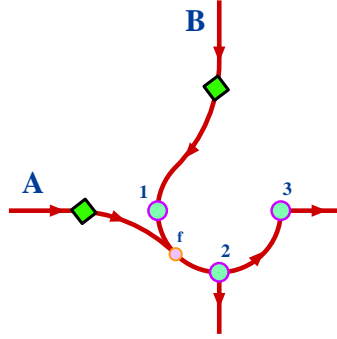


Figure 2 *Implementation scheme for the round-about.*

Now, this strategy requires the cellular automaton knows how to count up to two. As in [9], we use the three auxiliary structures which are represented in Figure 2 by a rhombus, a small circle and a bit larger ones.

These structures are : the fixed switch, see Sub-subsection 2.2.1, the doubler and the selector, see Sub-subsection 2.2.2. When the locomotive arrives close to the round-about, it first meets the doubler: it transforms the simple locomotive into a double one which consists of two contiguous black cells occupying blank cells of the tracks. Then, the locomotive may arrive at the fixed switch, depending on whether it arrived from **A** or from **B**. Then, the double locomotive arrives at the first selector: the structure recognizes a double locomotive. It kills one of the cells and the surviving simple locomotive is sent further on the round-about. When it meets the second selector, the structure recognizes a simple locomotive. Accordingly, it sends it on the track which leaves the round-about at that point.

In order to use the illustrations of this section and of the next one, we make use of a numbering of the tiles based on what is indicated by Figure 3. The central tile is numbered 0. Its sides are increasingly and counter-clockwise numbered from 1 to 5. Side 1 is defined as follows. Let V be the upper vertex of tile 0 which is on the vertical diameter δ of the right-hand side figure. The side 1 of tile 0 is the tile which shares V with tile 0 and which is on the left-hand side of δ .

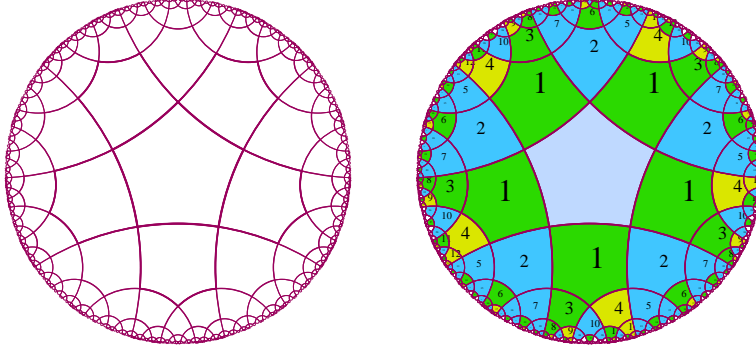


Figure 3 To left, the pentagrid. To right, the numbering of the tiles of the pentagrid.

Each side of tile 0 delimits a sector: sector 1 is headed by the tile 1(1) obtained from tile 0 by a reflection on its side 1. The sector is delimited by the rays starting from the other end of the side 1 of tile 0 which belong to edges of tile 1(1). The other sectors are obtained by appropriate rotations around the centre of tile 0 in such a way that the union of the sectors is the complement of tile 0 in the tiling. Inside each sector, the tiles are numbered as indicated in the right-hand side part of Figure 3. The tile ν of sector σ is numbered $\nu(\sigma)$. Each sector is panned by the same tree. In each tile, the sides are increasingly and counter-clockwise numbered from 1 to 5. The side 1 of the tile $\nu(\sigma)$ is the side it shares with its father in the tree rooted in the tile 1(σ). In Section 3, we shall indicate another way to number the sides of the tiles based on this numbering too. When we shall consider the cellular automaton constructed for the proof of Theorem 1, we shall speak of cell $\nu(\sigma)$ instead of tile $\nu(\sigma)$. In this notation, tile 0 will be denoted by 0(0). Once the side 1 is fixed in a cell c , its edges are numbered according to what we mentioned. The neighbour of c which shares the side i with it is called neighbour i . As we also consider as neighbours of c the cells which share a vertex only with c , we call these neighbours **vertex-neighbours** and we number them as follows: the vertices are increasingly and counter-clockwise numbered from 6 to 10, vertex 6 being shared by sides 1 and 5. Accordingly, the vertex-neighbour of c sharing its vertex j will be called neighbour j or vertex-neighbour j .

2.2.1 The fixed switch

As the tracks are one-way and as an active fixed switch always sends the locomotive in the same direction, there is no need of the other direction: there is no active fixed switch. Now, passive fixed switches are still needed as just seen in the previous paragraph. Figure 4 illustrates the passive fixed switch when there is no locomotive around. We can see that it consists of elements of the tracks which are simply assembled in the appropriate way in order to drive the locomotive to the bottom direction in the picture, whatever upper side the locomotive arrived at the switch. From our description of the working of the round-about, a passive fixed switch must be crossed by a double locomotive as well as a simple one.

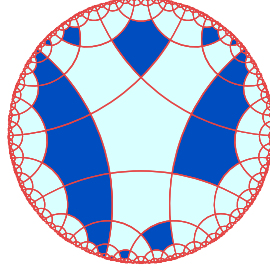


Figure 4 *The passive fixed switch in the pentagrid.*

Later, in Section 3.2, we shall check that the structure illustrated by Figure 4 allows these crossings. Note that the cell $0(0)$ in Figure 4 has six black neighbours instead of five of them in an ordinary cell of the tracks in a vertical track, see Figures 10 and 11, further. Its black neighbours are neighbours 2, 4, 6, 7, 8 and 10. Note that neighbour 10 is a milestone belonging to the cell $1(5)$ which is an element of the second track which arrives to the centre of the switch.

2.2.2 The doubler and the selector

The doubler is illustrated by the left-hand side picture of Figure 5. Note that the neighbourhood of the cell $0(0)$ is different from an ordinary element of the tracks and it is also different from the cell $0(0)$ of a fixed switch. Here, the black neighbours are neighbours 2, 4, 5, 7 and 8. It is slightly simpler than the doubler in [9] although we have two states only. As will be seen in Section 3, this is obtained by a particular numbering of the sides of a cell. The working of the configuration is the same as in [9], with this difference that the locomotive is black.

The selector is illustrated by the right-hand side of Figure 5. It is a very symmetric picture, rather different from that for the selector in [9]. In fact, the symmetry of the picture and an appropriate choice of the sides 1 allow us to make the cells $1(2)$ and $1(4)$ to react in a different way according to the number of black cells in the locomotive. Whatever this number, when the first cell of

the locomotive is in cell $0(0)$, both cells $1(2)$ and $1(4)$ can see whether or not a second black cell follows the first one in the cell $1(3)$. If a single black cell occurs, the cell $1(4)$ becomes black and the cell $1(2)$ remains white. If a second black cell occurs, the cell $1(4)$ remains white and the cell $1(2)$ turns to black.

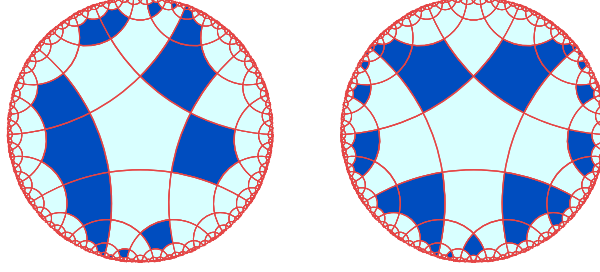


Figure 5 To left: the doubler. To right: selector.

After this point, in both cases, the configuration is that of tracks so the locomotive is sent on the appropriate path.

2.3 The fork, the controller and the controller-sensor

In this Sub-section, we look at the decomposition of two active switches: the flip-flop and the active part of the memory switch. We follow the same implementation as in [9], separating the working of the switch into two separate stages. The fork is implemented as in [9], but the controller is a bit different. For the convenience of the reader, we reproduce the figures of [9] which are again used by our automaton. In particular, we reproduce Figure 6. Figure 7 is different: the fork has the same configuration, but the two possible stable configurations of the controller are different, distinguished by the colour of the cell $1(1)$, especially when the controller has to stop the locomotive.

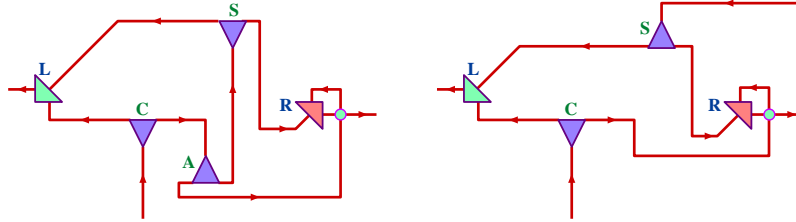


Figure 6 To left: the flip-flop. To right: the active memory switch.

When the simple locomotive arrives at the fork C , it is duplicated into two simple locomotives, each one following its own path. In the active memory switch, each locomotive goes to a controller. In the flip-flop, one of these locomotives goes on its way to the controller, and the other goes to another fork A

which sends one locomotive to the other controller and the other, which is now a third locomotive, is sent to a third fork S .

The locomotives sent by S go both to a controller, one to the black controller, the other to the white one. Now, when a locomotive sent by S arrives at a controller, it changes the black one into a white one and the white one into a black one. Accordingly, what should be performed by a flip-flop is indeed performed. It is enough to manage things in such a way that the locomotives arriving to the controllers from S arrive later than those sent by C and by A . In the active memory switch, the locomotive which arrives to S is sent by the passive memory switch. In this sense, the passive memory switch is active while the active one is passive.

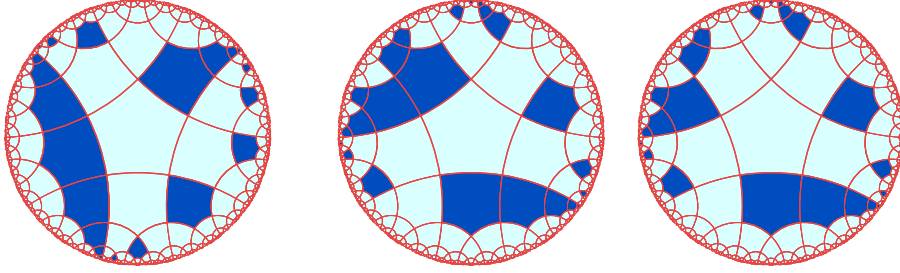


Figure 7 *To left: the fork. To right, two configurations: the controller of the flip-flop and of the active memory switch.*

In the fork, the locomotive, in its simple form, arrives through the cell 1(3). From the cell 0(0), two simple locomotives arise at the same time at the cells 1(1) and 1(4).

In the controller, the locomotive arrives through the path defined by the cells 21(4), 8(4), 3(4) and 1(4), in this order. These cells constitute a branch of the tree spanning the sector. Along this path, the simple locomotive arrives at cell 0(0) through the cell 1(4). When the cell 1(1) is black, it goes on its way along the track through the cells 1(2), 3(2), 8(2) and 21(2). When the cell 1(1) is white, the locomotive is stopped at the cell 1(4) and does not reach cell 0(0).

Figure 8 illustrates the construction of the passive memory switch with the help of forks and controllers. That implementation is somehow different from the one indicated in [9]. However, as in [9], the controllers of Figure 8 are not the same as those of Figure 6 and they are different from those of [9]. The main reason is that in the case of a passage of the locomotive through the non-selected track, in [9], the controller let the locomotive go, it changes the selection and it sends a signal to the other controller in order to order it to change its selection too. So that the controller has to perform three tasks at once. The present idea is to slightly simplify the task of the controller. As the locomotive should anyway not be stopped, we can place a fork on the tracks passively arriving to the switch: this is the reason for placing the forks S_1 and S_2 in Figure 8 where they are on the picture. Let us look at the locomotive which arrives at the fork S_2 on the figure. It corresponds to the non-selected track. The simple

Consider the case when the simple locomotive arrives to S_1 , the fork which corresponds on the figure to the selected track. Then, the locomotive is sent to L where it is stopped as no change should be performed.

8

It should be stressed that when the cell 1(1) is black, and when the locomotive arrives through the selected track, no locomotive is sent to the other controller as there is nothing to change.

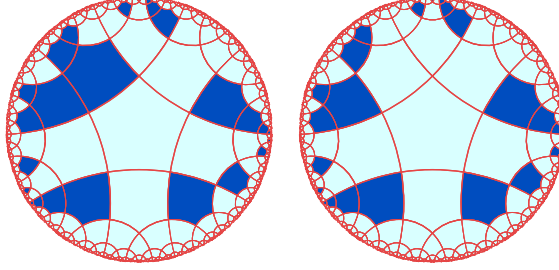


Figure 9 *The controller-sensor of the passive memory switch.*

3 Rules and figures

The figures of Section 2 help us to establish the rules. The rules and the figures were established with the help of a computer program which checked the coherence of the rules. The program also wrote the PostScript files of the pictures from the computation of the application of the rules to the configurations of the various type of parts of the circuit. The computer program also established the traces of execution which allow the reader to check the application of the rules.

We have to revisit the format of the rules and also to explain what is allowed from the relaxation from rotation invariance. We remind the reader that a rule has the form $\underline{x}_o x_1 \dots x_{12} \underline{x}_n$, where x_o is the state of the cell c , x_i is the state of the neighbour i of c and x_n is the new state of c once the rule has been applied. As the rules no more observe the rotation invariance, we may freely choose which is side 1 for each cell. We take this freedom from the format of the rule which only requires to know which is neighbour 1. In order to restrict the number of rules, it is decided that as a general rule, for a cell which is an element of the track, side 1 is the side shared by the cell and its next neighbour on the track. There can be exceptions when the cell is in a switch or the neighbour of the central cell in a switch. In particular, when a cell belongs to two tracks, side 1 is arbitrarily chosen among the two possible cases. The milestones have their side 1 shared by an element of the track. This means that when a milestone μ is a vertex-neighbour of an element of the track η , the side 1 of μ is not a side of η .

We have to keep in mind that there are two types of rules. Those which keep the structure invariant when the locomotive is far from them, we call this type of rules **conservative**, and those which control the motion of the locomotive. These latter rules, which we call **motion rules**, are the rules applied to the cells of the tracks as well as their milestones and, sometimes to the cells of the structures which may be affected by the passage of the locomotive.

Table 1 provides us with rules which concern the white cells which are not on the track and who have a single or two black neighbours. It also concerns black cells which have a single or two non white neighbours. A few other conservative rules of the same kind also appear in further tables.

Table 1 *Conservative rules.*

1	<u>wwwwwwwwww</u>	5	<u>wwwwbwbwww</u>	8	<u>bbwwwwwwwb</u>
2	<u>bwwwwwwwwb</u>	6	<u>wwwwbbbwwww</u>	9	<u>wbwwbwbwww</u>
3	<u>wbwwwwwwww</u>	7	<u>wwwwwbwww</u>	10	<u>wbbwwbwww</u>
4	<u>wwwwwbwww</u>				

3.1 Defining tracks and their rules

As explained in Section 2, the construction of the tracks is very important. The figures we give in this sub-section and in the following ones are *local views*. By that phrase, we mean that we can only see a very tiny part of the hyperbolic plane, that the central cell is not the centre of the hyperbolic plane, such a centre does not exist, but the cell on which we focus our attention. In these figures, we can see the central cell, an element of the tracks in Figures 10 and 11, as well as two other cells of the track, on both sides of the central cell.

We need to define verticals and horizontals in the pentagrid in order to simulate the corresponding Euclidean notions, which will allow us to directly implement the basic elements of the railway model, see [5, 10] for references. We look at verticals in Sub-subsection 3.1.1 and we look at horizontals in Sub-subsection 3.1.2, defining in these sub-subsections what we call verticals and horizontals in the pentagrid.

3.1.1 The vertical tracks

We call **line of the pentagrid** a line of the hyperbolic plane which supports a side of a tile of the pentagrid. Note that if a line δ of the hyperbolic plane supports a side s of some tile τ , it also supports the sides of a sequence $\{\tau_n\}_{n \in \mathbb{Z}}$ where $\tau = \tau_k$ for some $k \in \mathbb{Z}$ and such that each τ_n has a side on δ , that τ_n and τ_{n+1} share an edge for all $n \in \mathbb{Z}$ and that all τ_n are on the same side of δ . Such a set of tiles is called a **tape**. It is easy to see that a line of the pentagrid δ defines two tapes: each one is in its own side of δ . Say that a ray is a **ray of the pentagrid** if it is contained in a line of the pentagrid and if it is issued from a vertex of a tile of the pentagrid. We obtain a **half-tape** by replacing the line of the pentagrid δ by a ray of the pentagrid in the above definition of a tape. The tile of a half-tape defined by the ray of the pentagrid ρ which contains the vertex from which ρ is issued is called the **head** of the half-tape.

We say that a **vertical** is a finite sequence of consecutive tiles belonging to a half-tape η and which also belong to a branch of the Fibonacci tree rooted at the head of η . We remind the reader that the Fibonacci tree is the tree which generates the restriction of the pentagrid to a sector as illustrated in Figure 3.

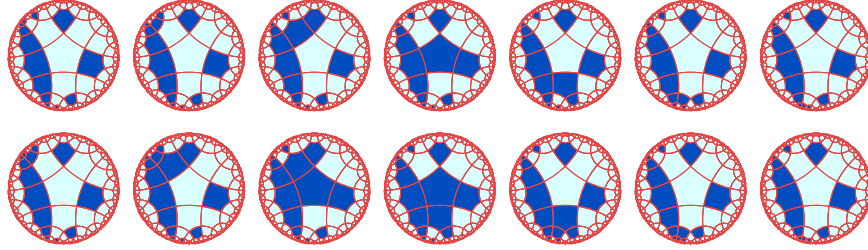


Figure 10 Above: a single locomotive going down a vertical track. Below: a double locomotive going down a vertical track.

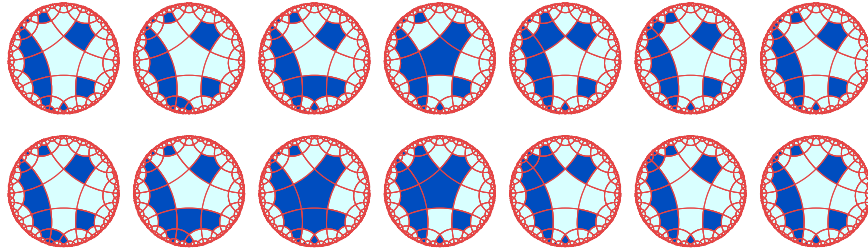


Figure 11 Above: a single locomotive going up a vertical track. Below: a double locomotive going up a vertical track.

A **vertical track** as illustrated by Figures 10 and 11 consists in two verticals which belong to half-tapes defined on both sides of the same ray of the pentagrid ρ . We say that ρ is the **backbone** of the vertical track. One vertical along ρ consists of blank cells while the other consists of black cells: each black cell β is a milestone of the white cell ω which shares a side on ρ with β . If the side 1 of ω is shared with the centre of the next element of the track, β is the neighbour 5 of ω while the second milestone of ω is its neighbour 2. Note that ω can see other milestones of its neighbouring elements of the track. In a going down track, neighbour 6 is a milestone in the next element of the tracks, it is also a neighbour of the neighbour 5 of ω . Neighbours 9 and 10 of ω are milestones of the previous element of the track, neighbour 10 being also a neighbour of the neighbour 5 of ω . When the track is going up, the vertex-neighbours are 7, a milestone of the next element of the track, and 8 with 9 which are milestones of the previous element.

Note that if we take the reflection in its backbone of the vertical track illustrated by Figure 10, we obtain the image of the vertical track illustrated by Figure 11 by a reflection in a vertex of a pentagon lying on the backbone of that vertical track. This shows us that Figures 10 and 11 illustrate all possible vertical track: going down as in Figure 10 or going up as in Figure 11 and again whichever the side where the track lies with respect to its backbone. This is

why the rules of Tables 2 and 4 also apply to tracks which are displayed on another side compared to that of Figures 10 and 11.

Let us now turn to the rules displayed by Table 2 for a locomotive which goes down and by Table 4 for a locomotive which goes up. When the locomotive is not there, the rule which is applied to the centre of an element of the track is rule 25 in Table 2: WWBWWBBWWBBW. Note that, as already mentioned, the milestones are neighbours 2 and 5 and that the milestones of neighbouring elements are seen by the centre as neighbours 6, 9 and 10 when the track goes down. When a simple locomotive crosses an element η in a vertical track, rule 26 makes the locomotive enter η . Indeed, the locomotive is seen by η as its neighbour 4. Rule 27 makes the locomotive go out: the cell turns from white to black. Rule 28 witnesses that the locomotive went out: it is seen as the neighbour 1 of η . Table 3 shows which rule are applied on the cells of the track around the central cell of Figure 10. We can see than the same sequence of rules: 25, 27, 28 and again 25 is applied along the cells of the tracks: 4(1), 1(1), 0(0), 1(3) and 3(3). These rules are motion rules as they apply to the centres of the elements of the tracks. They allow the locomotive to move on its way. But other rules are also needed. We can see there necessity in Table 3 which displays the rules applied to the cells 1(2) and 1(4) which are the milestones of the cell 0(0). When the locomotive is far from the element of the track, rule 12 applies to 1(2) and rule 2 applies to 1(4). For that latter cell it is clear: all neighbours of 1(4) are white. For the cell 1(2), it can see two other milestones on the half-line of black cells to which it belongs: its neighbour 2, a milestone of the cell of the track before 0(0) and its neighbour 5, a milestone of the cell of the track after 0(0). This is exactly what rule 12 says: with such a neighbouring, the cell remains blank. When the locomotive approaches, the milestone can first see it as its neighbour 7: rule 15 applies. Then, the milestone can see the locomotive as its neighbour 1: rule 16 applies. Then, the locomotive is going further, it is seen as neighbour 6: rule 17 applies. After that time, the locomotive is no more seen before a possible new passage on this very track, rule 12 applies again. All these rules: 12, 15, 16 and 17 leave the milestone unchanged. For the cell 1(4), other rules are applied with the same effect: the state is not changed. Rule 8 is applied when the locomotive is in the cell 0(0) and not before: it is seen as neighbour 1 of the milestone. Then rule 11 is applied, indicating that the locomotive, seen as neighbour 7, has left the cell 0(0). It is no more seen so that rule 2 again is applied.

Table 2 also displays the rules when the locomotive is double. Table 3 also indicates which rules are applied to the same cells as those seen previously. To some of the rules already examined, new rules are applied due to the fact that now two contiguous black cells travel together. When the locomotive approaches the cell 0(0), the same rule 26 is applied. But at the next time, instead of rule 27, rule 29 is applied: this rule makes the second black cell enter the cell 0(0) as this second cell is seen as the neighbour 4 of 0(0). Then, rule 31 is applied which makes the second black cell leave 0(0) because the first cell of the locomotive is seen as the neighbour 1 of 0(0). When the cell 0(0) is again white, rule 28 is again applied: it can see the second cell of the locomotive in neighbour 1. After that time, rule 25 is again applied. For the milestones, we have a similar

situation. For 1(2), after rule 15, rule 33 is applied: it can see the first cell of the locomotive in its neighbour 1 and the second one in its neighbour 7. Then rule 34 is applied which can see the first cell of the locomotive in the neighbour 6 of the milestone and the second cell of the locomotive in its neighbour 1. After that, rule 17 is applied as the first cell of the locomotive is no more seen by the milestone. For the cell 1(4), similarly, rule 32 is applied after rule 8 as two cells of the locomotive are seen: in the neighbour 7 of 1(4) and in its neighbour 1. Then rule 11 applies as a single cell of the locomotive can be seen. after that, rule 2 is again applied.

Table 2 *Rules forgoing down vertical tracks.*

a single locomotive:			
11	<u>B</u> W W W W W B W W W B	16	<u>B</u> B B W W B W W W W W B
12	<u>B</u> W B W W B W W W W W B	17	<u>B</u> W B W W B B W W W W W B
13	<u>W</u> B W W W W B W W W W W	18	<u>B</u> W W W W B B W W W W W B
14	<u>W</u> B W W W W B W W W W W	19	<u>B</u> W W W W B W W W W W W B
15	<u>B</u> W B W W B W B W W W B	20	<u>W</u> B W W W B W W W B B W
21	<u>W</u> W B W B B W W W B B W	25	<u>W</u> W B W W B B W W B B W
22	<u>B</u> B W W W W B W W W W B	26	<u>W</u> W B W B B B W W B B B
23	<u>W</u> W W W W W B B W W W W	27	<u>B</u> W B W W B B W W B B W
24	<u>W</u> W B W W B B W W W W W	28	<u>W</u> B B W W B B W W B B W
a double locomotive:			
29	<u>B</u> W B W B B B W W B B B	32	<u>B</u> B W W W W W B W W W B
30	<u>W</u> B W W W B B B W W W W	33	<u>B</u> B B W W B W B W W W B
31	<u>B</u> B B W W B B W W B B W	34	<u>B</u> B B W W B B W W W W B

Table 3 *Locomotive going down a vertical.*

	simple locomotive:						double locomotive:					
	0	1	2	3	4	5	0	1	2	3	4	5
4(1)	26	27	28	25	25	25	29	31	28	25	25	25
1(1)	25	26	27	28	25	25	26	29	31	28	25	25
0(0)	25	25	26	27	28	25	25	26	29	31	28	25
1(3)	25	25	25	26	27	28	25	25	26	29	31	28
3(3)	25	25	25	25	26	27	25	25	25	26	29	31
1(2)	12	12	15	16	17	12	12	15	33	34	17	12
1(4)	2	2	2	8	11	2	2	2	8	32	11	2

In Table 4, we have the rules for going up. Table 5 shows the rules which where applied in a motion illustrated by Figure 11, where the locomotives are going up. This time, rules 35, 41, 42 and 48 are used. Indeed, although side 1 is defined in the same way as in the case of Figure 10, the neighbourhood of the cell 0(0) is not exactly the same. This, time, as already mentioned, the milestones of the cell are still its neighbours 2 and 5, but the milestones of neighbouring elements are now seen in the neighbours 7, 8 and 9 instead of the neighbours 6, 9 and 10 in the previous case. That explains the differences we already noticed for a simple locomotive. Note that for the milestones, the rules are the same but they are applied in the reverse order: this is explained by the symmetry we already noticed.

Table 4 *Rules for going up vertical tracks.*
a single locomotive:

35 WWBWBWBWBWB
36 WWBWBWBWBWB
37 WWBWBWBWBWB
38 WWBWBWBWBWB
39 WWBWBWBWBWB
40 WWBWBWBWBWB

41 WBBWBWBWBWB
42 BWBWBWBWBWB
43 WWBWBWBWBWB
44 WBBWBWBWBWB
45 WBBWBWBWBWB
46 BWBWBWBWBWB

47 BWBWBWBWBWB
48 WBBWBWBWBWB
49 WWBWBWBWBWB
50 BWBWBWBWBWB
51 WBBWBWBWBWB

a double one:

52 BBBWBWBWBWB
53 BBBWBWBWBWB
54 WBBWBWBWBWB

Table 5 *Locomotive going up a vertical.*

	simple locomotive:						double locomotive:					
	0	1	2	3	4	5	0	1	2	3	4	5
3(3)	41	42	48	35	35	35	52	53	48	35	35	35
1(3)	35	41	42	48	35	35	41	52	53	48	35	35
0(0)	35	35	41	42	48	35	35	41	52	53	48	35
1(1)	35	35	35	41	42	48	35	35	41	52	53	48
4(1)	35	35	35	35	41	42	35	35	35	41	52	53
1(2)	12	12	17	16	15	12	12	17	34	33	15	12
1(5)	2	2	2	8	50	2	2	2	8	22	50	2

3.1.2 The horizontal tracks

We now investigate the construction of the horizontal tracks. For this, we have to define what we call **horizontals of the pentagrid**. These objects are not lines nor rays, they are sets of tiles defined with the help of the Fibonacci tree. A **level** of the Fibonacci tree is the set of nodes which are at the same distance from its root: this distance is the number of nodes on the branch going from the root to the node, the root being not taken into account. A **horizontal of the pentagrid** is a finite sequence of tiles whose image on in the Fibonacci tree belong to the same level of the tree, with this constraint, that two consecutive tiles in the sequence share a vertex. A **horizontal track** η is defined from a horizontal of the pentagrid h which is called the **backbone** of η .

Figure 12 shows us two possible configurations of a horizontal track depending on whether the tile around which they turn corresponds to a black tile or to a white one. Moreover, the figure is split into two parts: to left, the track goes in a clockwise way, to right, it goes on a counter-clockwise one.

Consider the case of a clockwise way around a black tile. The track consists of the following cells: 5(4), 4(3), 10(3), 3(3), 7(3), 2(3), 1(2), 2(2), 5(2), 4(1), 10(1), 3(1), 7(1), 18(1), 6(1), 15(1) and 5(1). Their 'isolated' milestones are, in the same order: 13(4), 11(3), 28(3), 26(3), 8(3), 20(3), 18(3), 5(3), 3(2), 6(2), 15(2), 13(2), 11(1), 28(1), 26(1), 8(1), 19(1), 49(1), 47(1), 16(1), 41(1), 39(1) and 13(1). The other milestones, those which share a side with another one, are: 2(1), 1(1), 0(0), 1(3) and 2(4). In our setting, we consider that 2(1) and 2(4) are white nodes and that 0(0) is a black one. As a consequence, 1(3) is a black node. If we assume that these three nodes belong to the same generation, we need to append the tiles 1(1) and 1(3) in order to avoid a cell of the track

with two consecutive neighbours numbered in 1..5. Indeed, in the same level of the tree, a black node is in between two white ones. The 'isolated' milestones are not always completely surrounded by blank cells. For instance, 19(1) shares a side with 41(1).

Similarly, the cells which constitute the clockwise horizontal track around a white cell are: 5(3), 4(2), 10(2), 3(2), 7(2), 2(2), 1(1), 2(1), 1(5), 2(5), 5(5), 4(4), 10(4), 3(4) and 7(4). The 'isolated' milestones are, in the same order: 13(3), 11(2), 28(2), 26(2), 8(2), 20(2), 18(2), 5(2), 3(1), 7(1), 5(1), 3(5), 6(5), 15(5), 13(5), 11(4), 28(4), 26(4), 8(4) and 19(4). The other milestones are, sharing at least a side with another one: 18(3), 6(3), 2(3), 1(2), 0(0), 1(3), 1(4), 2(4), 5(4) and 4(3). Note that 1(3) is almost surrounded by black cells. We assume that 2(4) is a black node, 0(0) and 2(3) are both white nodes, all of them the sons of a white node, here 1(3). In our case, 1(2) and 1(4) cannot be the centres of elements of the tracks so that we define these cells as milestones. This is a general feature: in between two consecutive nodes ν_1 and ν_2 on the same level which are milestones, the black node which shares a side with both ν_1 and ν_2 must also be a milestone.

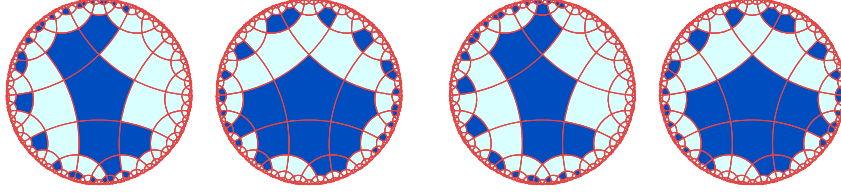


Figure 12 *To left: a clockwise horizontal track around a node: first, around a black node and then, around a white one. To right: a counter-clockwise horizontal track around a node: first, around a black node and then, around a white one.*

For the counter-clockwise tracks, the centres of the elements of the track are the same as in the clockwise case. The changes occur in the 'isolated' milestones. For the black node track, 5(4), 11(3), 8(3), 5(3), 3(2), 6(2), 11(1), 8(1), 19(1), 16(1) and 13(1) are replaced by 6(4), 12(3), 9(3), 6(3), 4(2), 7(2), 12(1), 9(1), 20(1), 17(1) and 14(1) respectively. Similarly, for the white node track, the milestones: 13(3), 11(2), 8(2), 5(2), 3(1), 3(5), 6(5), 11(4), 8(4) and 19(4) are replaced by 14(3), 12(2), 9(2), 6(2), 4(1), 4(5), 7(5), 12(4), 9(4) and 20(4) respectively.

Tables 6 and 7 display the rules for the motion on a horizontal in a clockwise way around a black node. The corresponding motion is illustrated by Figure 13. Also, Tables 8 and 9 show the application of the rules of tables 6 and 7 respectively, on cells of horizontal tracks and on some of their milestones. The motion is illustrated by Figure 14.

We can see that the centres of the elements of the tracks are applied different rules from those we have seen previously, although horizontal tracks consists of the same elements of the tracks. Let us look at this more carefully. For instance, in Figure 13, when it is visited by a simple locomotive, the cell 3(1)

is applied rules 58, 78, 83, 87 and 90 respectively. Rule 58, WBWBBWBWB, is a conservative rule: it leaves the cell blank and it is applied when the locomotive is not by the cell. Note that the milestones are in neighbours 2 and 5, but the cell can see the milestones of neighbouring cells in its vertex-neighbours 6, 8 and 9, a very slight difference with rule 35. When the locomotive approaches 3(1), it is seen in its neighbour 3 and rule 78, WBWBWBWBWB, is applied. Then, rule 83, BWBWBWBWBWB, is applied which makes the locomotive leave the cell. Next, rule 87, WBWBWBWBWBWB, is applied: it witnesses that the locomotive is now in neighbour 1. Then, rule 90, WBWBWBWBWBWB, witnesses that the locomotive is in the vertex-number 6 of 3(1). Note that for the cell 4(1), the neighbourhood is a bit different. Besides the milestones of the element in neighbours 2 and 5, other milestones, from neighbouring elements, can be seen in the vertex-neighbours 6 and 9 only. Now, rule 58 again apply as the locomotive is seen in the vertex-neighbour of the cell. Later, rule 88, WBWBWBWBWBWB, makes the locomotive enter the cell as it is seen in its neighbour 3. Rule 61, BWBWBWBWBWB, makes it leave the cell, rule 71, WBWBWBWBWBWB, can see it in its neighbour 1 and rule 79, WBWBWBWBWBWB, can see it in its vertex-neighbour 7.

Table 6 Rules for a single locomotive, counter-clockwise moving on a horizontal track, here, around the sons of a black node.

55	<u>B</u> BWBWBWBWBW <u>B</u>	66	<u>W</u> BWBWBWBWBW <u>B</u>	77	<u>B</u> BBWWWWBWBW <u>B</u>	87	<u>W</u> BBWBWBWBWBW <u>B</u>
56	<u>B</u> WWBBWWWWWW <u>B</u>	67	<u>W</u> BBWWWWBWBW <u>B</u>	78	<u>W</u> BBWBWBWBWBW <u>B</u>	88	<u>W</u> BBWBWBWBWBW <u>B</u>
57	<u>B</u> WBWBWBWBWBW <u>B</u>	68	<u>W</u> WWWWWWBWBW <u>B</u>	79	<u>W</u> BWBWBWBWBWBW <u>B</u>	89	<u>B</u> BWBWBWBWBWBW <u>B</u>
58	<u>W</u> BWBWBWBWBWBW <u>B</u>	69	<u>W</u> BWBWBWBWBWBW <u>B</u>	80	<u>B</u> WBWBWBWBWBW <u>B</u>	90	<u>W</u> BWBWBWBWBWBW <u>B</u>
59	<u>W</u> WBWBWBWBWBW <u>B</u>	70	<u>B</u> WBWBWBWBWBW <u>B</u>	81	<u>B</u> WWBBBWWWWWW <u>B</u>	91	<u>B</u> WWBBWBWBWBW <u>B</u>
60	<u>W</u> BWBWBWBWBWBW <u>B</u>	71	<u>W</u> BBWBWBWBWBW <u>B</u>	82	<u>B</u> WBWBWBWBWBW <u>B</u>	92	<u>B</u> BWBWBWBWBWBW <u>B</u>
61	<u>B</u> WBWBWBWBWBW <u>B</u>	72	<u>W</u> BBWBWBWBWBW <u>B</u>	83	<u>B</u> WBWBWBWBWBW <u>B</u>	93	<u>B</u> WBWBWBWBWBW <u>B</u>
62	<u>W</u> WBWBWBWBWBW <u>B</u>	73	<u>B</u> WBWBWBWBWBW <u>B</u>	84	<u>W</u> BBWBWBWBWBW <u>B</u>	94	<u>B</u> BBWBWBWBWBW <u>B</u>
63	<u>W</u> BWBWBWBWBWBW <u>B</u>	74	<u>B</u> BBWWWWWWWW <u>B</u>	85	<u>B</u> WBWBWBWBWBW <u>B</u>	95	<u>B</u> WBWBWBWBWBW <u>B</u>
64	<u>B</u> BWBWBWBWBWBW <u>B</u>	75	<u>W</u> BWBWBWBWBWBW <u>B</u>	86	<u>B</u> WBWBWBWBWBW <u>B</u>	96	<u>B</u> BWBWBWBWBWBW <u>B</u>
65	<u>W</u> BBWBWBWBWBW <u>B</u>	76	<u>B</u> WWBBBWWWWWW <u>B</u>				

Table 7 Rules for a double locomotive, counter-clockwise moving on a horizontal track, here, around the sons of a black node.

97	<u>B</u> WBWBWBWBWBW <u>B</u>	103	<u>W</u> BBWBWBWBWBW <u>B</u>	109	<u>B</u> BBWBWBWBWBW <u>B</u>	115	<u>B</u> BWBWBWBWBWBW <u>B</u>
98	<u>B</u> BBWBWBWBWBW <u>B</u>	104	<u>B</u> WBWBWBWBWBW <u>B</u>	110	<u>B</u> WBWBWBWBWBW <u>B</u>	116	<u>B</u> WBWBWBWBWBW <u>B</u>
99	<u>B</u> BWBWBWBWBWBW <u>B</u>	105	<u>B</u> BWBWBWBWBWBW <u>B</u>	111	<u>B</u> BWBWBWBWBWBW <u>B</u>	117	<u>B</u> BBWBWBWBWBW <u>B</u>
100	<u>B</u> WBWBWBWBWBW <u>B</u>	106	<u>B</u> WBWBWBWBWBW <u>B</u>	112	<u>B</u> BWBWBWBWBWBW <u>B</u>	118	<u>B</u> WBWBWBWBWBW <u>B</u>
101	<u>B</u> BBWBWBWBWBW <u>B</u>	107	<u>B</u> BBWBWBWBWBW <u>B</u>	113	<u>W</u> BBWBWBWBWBW <u>B</u>	119	<u>B</u> BBWBWBWBWBW <u>B</u>
102	<u>B</u> BBWBWBWBWBW <u>B</u>	108	<u>B</u> WBWBWBWBWBW <u>B</u>	114	<u>B</u> WBWBWBWBWBW <u>B</u>		

Now, the cells 10(1) and 5(2) are applied the same rules as they are in the same situation, playing the same role. The locomotive enters such a cell in a standard way but it leaves the cell from an entry and not from the side which is in between two ones which are shared by a black neighbour. This can be performed in one direction and also in the opposite one as seen later, with Table 10. Rule 62, WWBWBWBWBWBWB, is the conservative rule: it can see the milestones in the neighbours 3, 5 and vertex-neighbours 7 and 8. The locomotive enters through side 2, rule 65, WWBWBWBWBWBWB, and it leaves the cell by rule 73, BWBWBWBWBWBWB. Rule 63, WBWBWBWBWBWBWB, can see the leaving locomotive in the neighbour 1 of the cell. The milestones 12(1) and 13(1) are applied already used rules in other contexts in the vertical motions. We leave this

study to the reader. We also do the same for the motions when the horizontal track goes through the sons of a white node: see Tables 13 and 16 for the rules, Tables 14, 15, 17 and 18 for the application of the rules to cells of the tracks and a few milestones which are taken from Figures 15, and 16.

Table 8 *Execution of the rules of Table 6 for a simple locomotive counter-clockwise running around the sons of a black node.*

	2	3	4	5	6	7	8
3(1)	78	83	87	90	58	58	58
10(1)	62	65	73	63	62	62	62
4(1)	59	58	88	61	71	79	59
5(2)	62	62	62	65	73	63	62
2(2)	66	66	66	35	72	80	84
12(1)	19	19	19	64	18	19	19
13(2)	46	46	46	47	74	46	46

Table 9 *Execution of the rules of Table 7 for a double locomotive counter-clockwise running around the sons of a black node.*

	2	3	4	5	6	7	8
3(1)	108	111	113	90	58	58	58
10(1)	65	100	105	63	62	62	62
4(1)	58	78	114	98	103	79	59
5(2)	62	62	65	100	105	63	62
2(2)	66	66	35	41	104	109	84
12(1)	19	19	64	99	18	19	19
13(2)	46	46	47	101	74	46	46

Table 10 *Execution of the rules of Table 12 for a simple locomotive clockwise running around a black node:*

simple locomotive:				double locomotive:			
	0	1	2	3	4	5	6
4(3)	123	129	132	121	121	121	121
10(3)	124	130	133	122	122	122	122
3(3)	121	128	80	84	79	66	66
7(3)	122	122	124	130	133	122	122
2(3)	59	59	25	135	61	71	59
8(3)	46	46	74	47	46	46	46
20(3)	19	19	18	64	19	19	19

Table 11 Execution of the rules of Table 12 for a double locomotive clockwise running around a black node:

	0	1	2	3	4	5	6
4(3)	141	145	132	121	121	121	121
10(3)	142	146	133	122	122	122	122
3(3)	137	144	109	103	79	66	66
7(3)	122	124	142	146	133	122	122
2(3)	59	25	26	149	98	71	59
8(3)	46	74	101	47	46	46	46
20(3)	19	18	99	64	19	19	19

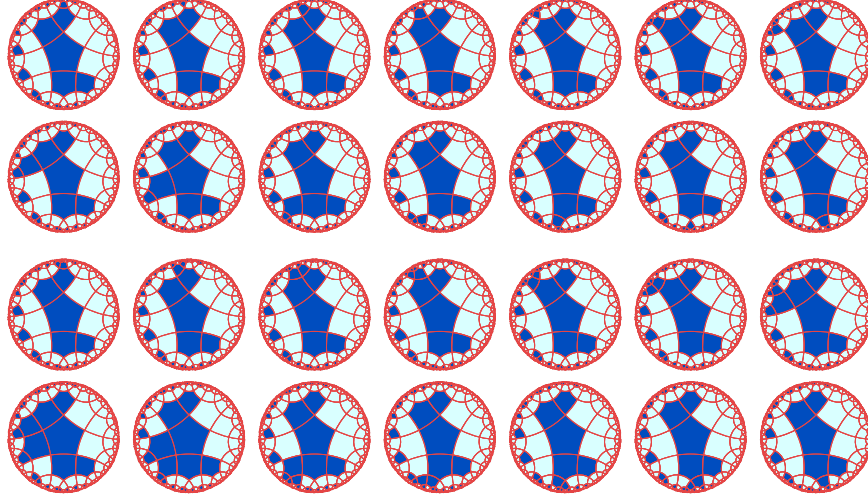


Figure 13 A locomotive *counter-clockwise* moving on a horizontal track, here around the sons of a **black node**. Above: a simple locomotive; below: a double one.

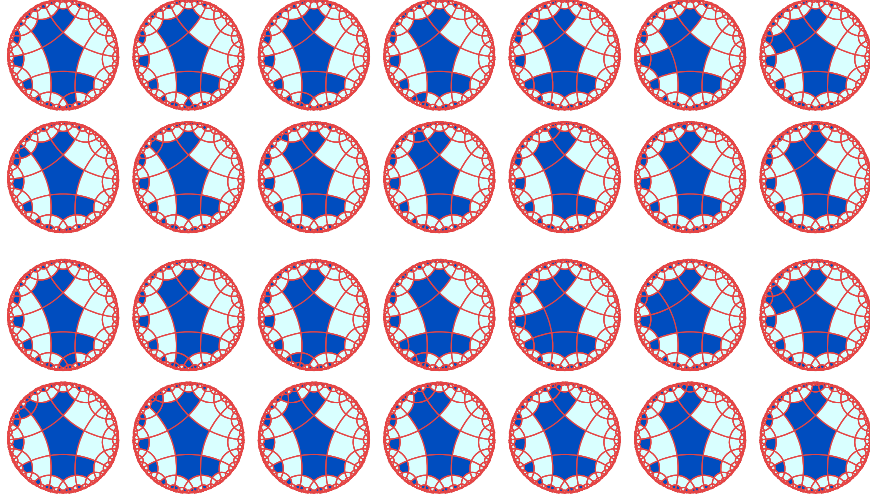


Figure 14 A locomotive *clockwise* moving on a horizontal track, here around the sons of a **black node**. Above: a simple locomotive; below: a double one.

In the abstract and in the introduction, we mentioned that we relaxed the condition on the rotation invariance of the cellular automaton constructed for the proof of Theorem 1. In this part of our study, we can see six rules which are concerned with this point. As an example: rule 21, $\underline{W}WBWBWBWBWB$, has a rotated neighbourhood which is also that of rule 65: $\underline{W}WBWBWBWBWB$. The state of the cell is the same, but the new state is different. Rule 21 appeared in the motion of a simple locomotive going down a vertical track, while rule 65 appears in the motion of a simple locomotive counter-clockwise moving along a horizontal track. A higher number of rules are the rotated form of another one. For instance, rules 81, 93 and 136 are in this case: rule 93, $\underline{B}WBWBWBWBWB$, and rule 136, $\underline{B}WBWBWBWBWB$, are both rotated forms of rule 81, $\underline{B}WBWBWBWBWB$, as can easily be seen. At the end of the paper we shall look at all rules which have the same neighbourhood, up to rotation, the same state but a different new state.

Table 12 Rules for the clockwise motion of a locomotive around a black node.
simple locomotive

120	$\underline{W}WBWBWBWBWB$	125	$\underline{W}WBWBWBWBWB$	130	$\underline{B}WBWBWBWBWB$	135	$\underline{W}WBWBWBWBWB$
121	$\underline{W}WBWBWBWBWB$	126	$\underline{W}WBWBWBWBWB$	131	$\underline{B}WBWBWBWBWB$	136	$\underline{B}WBWBWBWBWB$
122	$\underline{W}WBWBWBWBWB$	127	$\underline{B}WBWBWBWBWB$	132	$\underline{W}WBWBWBWBWB$	137	$\underline{W}WBWBWBWBWB$
123	$\underline{B}WBWBWBWBWB$	128	$\underline{W}WBWBWBWBWB$	133	$\underline{B}WBWBWBWBWB$	138	$\underline{B}WBWBWBWBWB$
124	$\underline{W}WBWBWBWBWB$	129	$\underline{B}WBWBWBWBWB$	134	$\underline{B}WBWBWBWBWB$	139	$\underline{B}WBWBWBWBWB$

double locomotive

140	$\underline{B}WBWBWBWBWB$	143	$\underline{B}WBWBWBWBWB$	146	$\underline{B}WBWBWBWBWB$	149	$\underline{B}WBWBWBWBWB$
141	$\underline{B}WBWBWBWBWB$	144	$\underline{B}WBWBWBWBWB$	147	$\underline{B}WBWBWBWBWB$	150	$\underline{B}WBWBWBWBWB$
142	$\underline{B}WBWBWBWBWB$	145	$\underline{B}WBWBWBWBWB$	148	$\underline{B}WBWBWBWBWB$	151	$\underline{B}WBWBWBWBWB$

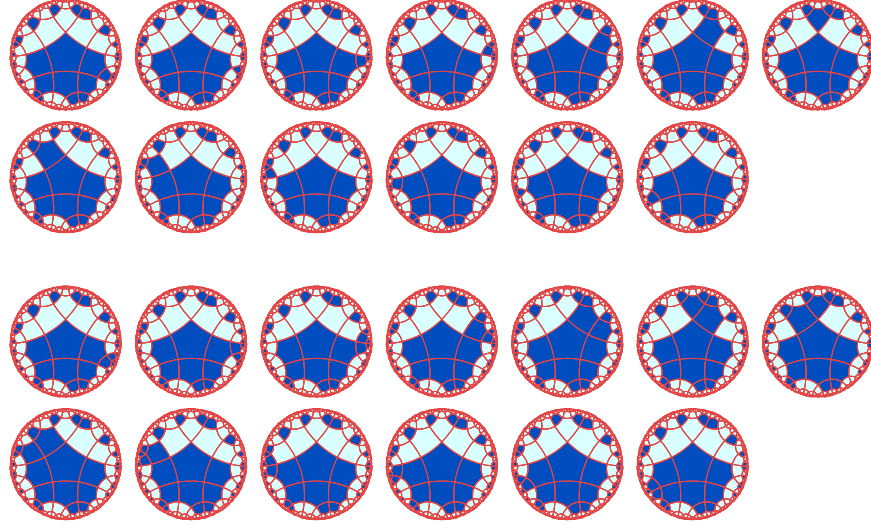


Figure 15 A locomotive *counter-clockwise* moving on a horizontal track, here around the sons of a *white node*. Above: a simple locomotive; below: a double one.

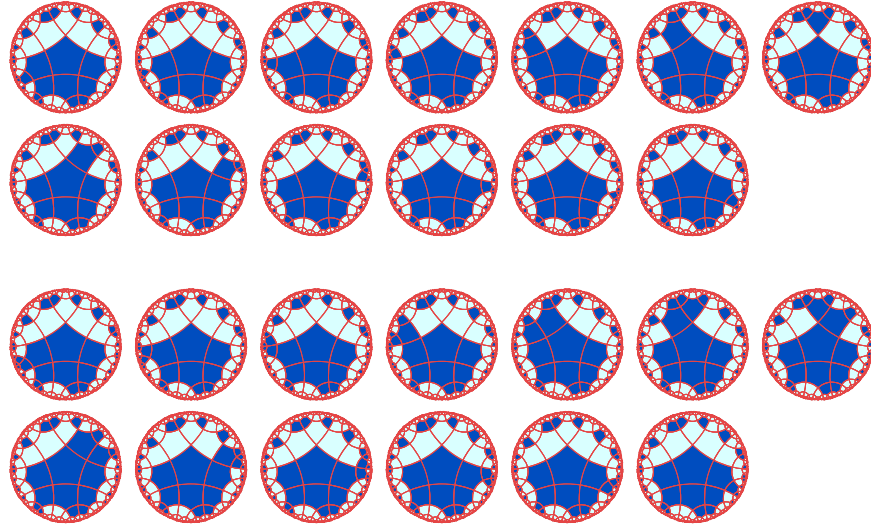


Figure 16 A locomotive *clockwise* moving on a horizontal track, here around the sons of a *white node*. Above: a simple locomotive; below: a double one.

Table 13 Rules for the counter-clockwise motion of a locomotive around a white node.

simple locomotive					
152	<u>B</u> BBBBWWWBB <u>B</u>	158	BBBBWWWBBW <u>B</u>	164	BBBBWWWBBW <u>B</u>
153	<u>B</u> WWWBBWWWBB <u>B</u>	159	<u>B</u> WBWWWWWBB <u>B</u>	165	<u>B</u> WBWWWWWBB <u>B</u>
154	<u>B</u> BBWBWWWWWBB	160	<u>B</u> BBWWWWWBBW <u>B</u>	166	<u>B</u> WBWWWWWBBW <u>B</u>
155	<u>B</u> WBWWWWWBBW <u>B</u>	161	<u>B</u> WBWWWWWBBW <u>B</u>	167	<u>B</u> WBWWWWWBBW <u>B</u>
156	<u>W</u> BBWWWWWBBW <u>W</u>	162	<u>B</u> WBWWWWWBBW <u>B</u>	168	<u>B</u> WBWWWWWBBW <u>B</u>
157	<u>W</u> BWWWWWBBW <u>W</u>	163	<u>B</u> WBWWWWWBBW <u>B</u>	169	<u>B</u> WBWWWWWBBW <u>B</u>
double locomotive					
176	<u>B</u> WBWWWWWBBW <u>B</u>	180	<u>B</u> WBWWWWWBBW <u>B</u>	183	<u>B</u> WBWWWWWBBW <u>B</u>
177	<u>B</u> WBWWWWWBBW <u>B</u>	181	<u>B</u> WBWWWWWBBW <u>B</u>	184	<u>B</u> WBWWWWWBBW <u>B</u>
178	<u>B</u> WBWWWWWBBW <u>B</u>	182	<u>B</u> WBWWWWWBBW <u>B</u>	185	<u>B</u> WBWWWWWBBW <u>B</u>
179	<u>B</u> WBWWWWWBBW <u>B</u>				

Table 14 Execution of the rules 13 for a simple locomotive counter-clockwise running around the sons of a white node.

	6	7	8	9	10	11
2(2)	58	78	83	87	90	58
7(2)	62	62	65	73	63	62
3(2)	59	59	58	88	61	71
10(2)	62	62	62	62	65	73
4(2)	66	66	66	66	35	72
1(2)	153	170	172	173	174	175
0(0)	167	169	171	152	152	152

Table 15 Execution of the rules 13 for a double locomotive counter-clockwise running around the sons of a white node.

	6	7	8	9	10	11
2(2)	78	108	111	113	90	58
7(2)	62	65	100	105	63	62
3(2)	59	58	78	114	98	103
10(2)	62	62	62	65	100	105
4(2)	66	66	66	35	41	104
1(2)	170	184	185	186	155	187
0(0)	182	183	171	152	152	152

Table 16 Rules for the clockwise motion of a locomotive around a white node.

189	<u>W</u> BBWWBBWW <u>B</u>	190	<u>W</u> BBWWBBWW <u>B</u>
-----	----------------------------	-----	----------------------------

Table 17 Execution of the rules 13 and 16 for a simple locomotive clockwise running around the sons of a white node.

	0	1	2	3	4	5	6
4(2)	123	129	132	121	121	121	121
10(2)	124	130	133	122	122	122	122
3(2)	121	128	80	84	79	66	66
7(2)	122	122	124	130	133	122	122
2(2)	59	59	25	135	61	71	59
1(2)	106	175	174	173	172	170	153
0(0)	152	152	152	152	171	169	167

Table 18 Execution of the rules 13 and 16 for a double locomotive clockwise running around the sons of a white node.

	0	1	2	3	4	5	6
4(2)	141	145	132	121	121	121	121
10(2)	142	146	133	122	122	122	122
3(2)	137	144	109	103	79	66	66
7(2)	122	124	142	146	133	122	122
2(2)	59	25	26	149	98	71	59
1(2)	187	155	186	185	184	170	153
0(0)	152	152	152	171	183	182	181

3.2 The fixed switch

Figures 17 and 18 illustrate how the locomotive passively crosses a fixed switch. Both figures illustrate the case when the locomotive comes from the left-hand side and when it comes from the right-hand side. Note that the locomotive may be simple as well as double: indeed, the fixed switch plays a role in the round-about, this is why it may be crossed by a double locomotive.

Table 19 displays the rules used for the passive fixed switch. More precisely, the table gives the rules which were not yet used. In fact several rules from the vertical motions are also used in this case.

Table 19 Rules for the passive fixed switch:

from left, simple one:

191	<u>W</u> WBWBBWBBB <u>W</u>
192	<u>W</u> WWWWWWBB <u>W</u>
193	<u>W</u> WBWBBWBBB <u>B</u>
194	<u>B</u> WBWBBWBBB <u>W</u>
195	<u>W</u> BBWBBWBBB <u>W</u>
196	<u>B</u> WWWWBWBWW <u>B</u>

from left, double one:

197	<u>B</u> WBWBBWBBB <u>B</u>
198	<u>B</u> BBWBBWBBB <u>W</u>
199	<u>B</u> BWWBWBWW <u>B</u>

from right, simple one:

200	<u>W</u> WBWBBWBBB <u>B</u>
-----	-----------------------------

from right, double one:

201	<u>B</u> WBWBBWBBB <u>B</u>
-----	-----------------------------

As illustrated by the figures, the switch is passive and it works both for a single and a double locomotive whatever the side from which they arrive to the

switch. Note that the exceptional cell is the centre of the switch, the cell 0(0) in the pictures of Figures 17 and 18. Rule 191, WWBWWBBWBBB, is the conservative rule which is applied when no locomotive crosses the switch. Its milestones are neighbours 2, and 5 and also the vertex-neighbours 6, 8, 9 and 10 which belong to milestones of neighbouring elements of the tracks. Note that the cell 2(1) is a milestone shared by two elements, each one belonging to one of the tracks which arrive to the switch.

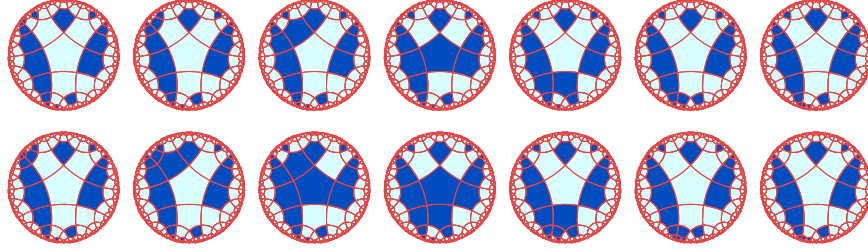


Figure 17 *Locomotives passively crossing a fixed switch from the left-hand side. Above : a single locomotive; below: a double one.*

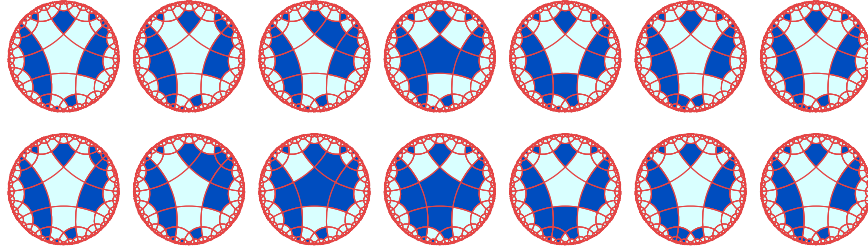


Figure 18 *Locomotives passively crossing a fixed switch from the right-hand side. Above : a single locomotive; below: a double one.*

Table 20 gives traces of executions for a simple locomotive crossing the switch, either from the left, left-hand side part of the table, or from the right, right-hand side part of the table. Table 21 does the same for the same cells when a double locomotive crosses the switch.

Table 20 *Execution of the rules on the cells 1(1), 1(5), 0(0), 1(3), 1(2) and 1(4) when a simple locomotive passively crosses the fixed switch.*

	from the left:						from the right:					
	0	1	2	3	4	5	0	1	2	3	4	5
1(1)	25	26	27	28	25	25	25	25	132	28	25	25
1(5)	35	35	90	48	35	35	35	41	42	48	35	35
0(0)	191	191	193	194	195	191	191	191	200	194	195	191
1(3)	25	25	25	26	27	28	25	25	25	26	27	28
1(2)	12	12	15	16	17	12	12	12	12	16	17	12
1(4)	19	19	19	64	196	19	19	19	18	64	196	19

Table 21 *Execution of the rules on the cells 1(1), 1(5), 0(0), 1(3), 1(2) and 1(4) when a double locomotive passively crosses the fixed switch.*

	from the left:						from the right:					
	0	1	2	3	4	5	0	1	2	3	4	5
1(1)	26	29	31	28	25	25	25	132	145	28	25	25
1(5)	35	90	113	48	35	35	41	52	53	48	35	35
0(0)	191	193	197	198	195	191	191	200	201	198	195	191
1(3)	25	25	26	29	31	28	25	25	26	29	31	28
1(2)	12	15	33	34	17	12	12	12	16	34	17	12
1(4)	19	19	64	199	196	19	19	18	99	199	196	19

These traces deal with the cells 1(1), 1(5), 0(0) and 1(3) for the tracks and with the cells 1(2) and 1(4) for the particular milestones which are not in the situation of milestones in a vertical track or around a set of cells.

Note that the side from which the locomotive arrives to the cell 0(0) is indicated by rule 193, wwBWBWBWB, for the left-hand one and rule 200, wwBWBWBWB, for the right hand one. For rule 193, the locomotive can be seen in the neighbour 4 of 0(0) when it comes from the left-hand side. For rule 200, it can be seen in neighbour 3 when it comes from the right-hand side. It can be checked that many rules from Table 2 are used in these executions, this is especially the case for the conservative rules applied to all cells of the tables. The motion rules are also those of the vertical tracks for cells 1(1) and 1(3).

3.3 The round-about

In Section 2, we have seen the idle configurations of the other pieces of the round-about: the doubler, see Sub-subsection 3.3.1 and the selector, see Sub-subsection 3.3.3.

3.3.1 The doubler

Figure 19 illustrates the motion of the locomotive through the doubler. The locomotive arrives in the form of a simple one. The creation of the second

cell of the locomotive happens in a simple way: when the locomotive enters the cell 1(1), at the same moment, a black cell is created in the cell 0(0). That event constitutes the creation of the double locomotive, which is clear in Figure 19 as well as in Table 23. The rules are given in Table 22. However, rules of previous tables are also involved, in particular rules from the motions on the vertical tracks.

The cells of the track involved in the doubler are, in the order they are traversed: 2(1), 1(1), 0(0), 1(3), 3(3) and 8(3). The milestones are in the cells: 4(5), 15(1), 1(5), 7(1), 3(1), 2(2), 1(2), 1(4), 2(3), 4(3), 7(3) and 9(3). Here, the conservative rule for the central cell is rule 202: wwbbwbbwwbw. We can see the milestones in neighbours 2, 3 and 5 as well as in vertex-neighbours 6 and 10. The creation of the double locomotive is performed by rules 207, wwbbwbbbwbb, and rule 206, wbbwbbwbbw, simultaneously applied to the cell 1(1) and to 0(0), respectively. Note that in rule 206, the occurrence of a black cell in the vertex-neighbour 9 triggers the creation of the front cell of the locomotive.

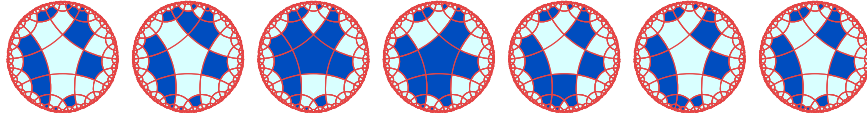


Figure 19 The structure which doubles a locomotive: a single locomotive enters the structure; a double one leaves it.

Table 22 Rules for the doubler.

202	<u>wwbbwbbwwbw</u>	206	<u>wbbwbbwbbw</u>	210	<u>bbwbbwbbwwb</u>	214	<u>bbwbbwbbwwb</u>
203	<u>wwbbwbbbwbb</u>	207	<u>wwbbwbbbwbb</u>	211	<u>bbbwbbwbbw</u>	215	<u>wbbwbbwbbw</u>
204	<u>bwwbbwbbwbb</u>	208	<u>bwbwbbwbbw</u>	212	<u>wbbwbbwbbw</u>	216	<u>bwwbbwbbwbb</u>
205	<u>wbbwbbwbbw</u>	209	<u>bbwbbwbbwbb</u>	213	<u>bbwbbwbbwwb</u>	217	<u>bwwbbwbbwbb</u>

Table 23 Execution of a run through the doubler.

	0	1	2	3	4	5
2(1)	78	83	113	90	58	58
1(1)	203	207	209	212	203	203
0(0)	202	206	208	211	215	202
3(1)	25	25	26	29	31	28
2(2)	19	19	99	18	19	19
1(2)	12	12	33	34	17	12
2(3)	12	12	15	33	34	17
1(5)	204	134	210	214	217	217
1(4)	50	50	22	213	216	50

3.3.2 The fork

Although the fork is not connected with the round-about, we place it here as it appears at that point in the table of the rules. However, it has some link with

the doubler as, like for that structure, two locomotives are issued from the fork. Now, in contrast with what happens in the doubler where both locomotives travel together, always in contact with one another, in the fork each created simple locomotive goes its own way: they travel in different directions. Figure 20 illustrates this structure. Table 24 gives the rules for the fork and Table 25 displays the rules which are applied in the visit of the fork by a locomotive.

The cells of the tracks involved in the structure are: 8(3), 3(3), 1(3), 0(0) and then 1(1), 4(1), 12(1) and 33(1) in one direction and 1(4), 2(5), 7(5), 20(5) and 54(5) in the other direction. The milestones are: 20(3), 23(3), 7(3), 10(3), 2(3), 2(4) and then 1(2), 1(5), 2(2), 3(1), 5(2), 11(1), 13(2) and 32(1) in the first direction and 4(4), 6(5), 3(5), 19(5), 8(5) and 53(5) for the second one.

Table 24 *Rules for the fork.*

218	<u>WWWBWBWBWWW</u>	221	<u>WBBWWWBWBW</u>	224	<u>BWBWBWBWBWB</u>	227	<u>BWBWBWBWBWB</u>
219	<u>BWBWBWBWBWB</u>	222	<u>WBBWBWBWBWB</u>	225	<u>WBBWBWBWBWB</u>	228	<u>WBBWBWBWBWB</u>
220	<u>WBWBWBWBWB</u>	223	<u>BWBWBWBWBWB</u>	226	<u>BWBWBWBWBWB</u>	229	<u>WBWBWBWBWB</u>

Table 25 *Execution of the rules of the fork.*

	0	1	2	3	4	5
1(3)	35	41	42	48	90	35
0(0)	35	35	41	42	225	228
1(1)	35	35	35	41	42	48
1(4)	218	218	62	222	226	229
2(5)	66	66	66	35	72	80
1(5)	219	219	219	223	227	224

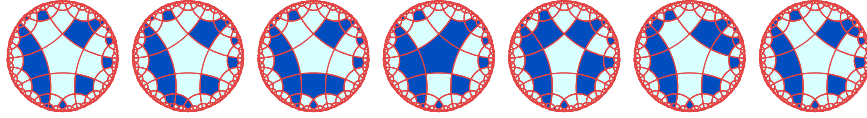


Figure 20 *The fork: a single locomotive enters the fork. Two single one leave it in different directions.*

The neighbourhood of the cell 0(0) is exactly that of a cell of going up vertical tracks. It is given by the conservative rule 35: WBBWBWBWBWB. The milestones are in neighbours 2 and 5 and the cell can see the milestones of the neighbouring elements in the vertex-neighbours 7, 8 and 9. When the simple locomotive is in the cell 0(0), rules 41, WBBWBWBWBWB, and rule 222, WBBWBWBWBWB, apply to cell 1(1) and 1(4) respectively, simultaneously creating a locomotive at the corresponding place. In the cell 1(1), the locomotive can be seen in its neighbour 3 while in the cell 1(4), it can be seen in its neighbour 2. Note that the cell 1(4) behaves like the cells 10(1) and 5(2) we have looked at in Sub-section 3.1.2. The cell 1(4) allows the locomotive to perform a turn at right-angle.

3.3.3 The selector

Figure 21 illustrates the working of the selector of the round-about. The case of a single locomotive is illustrated by the first two rows of the figure while the case of a double locomotive is illustrated by the last row. Tables 26 and 27 give the rules for the locomotive and for the cells of the selector, Table 26 for a simple locomotive, Table 27 for a double one.

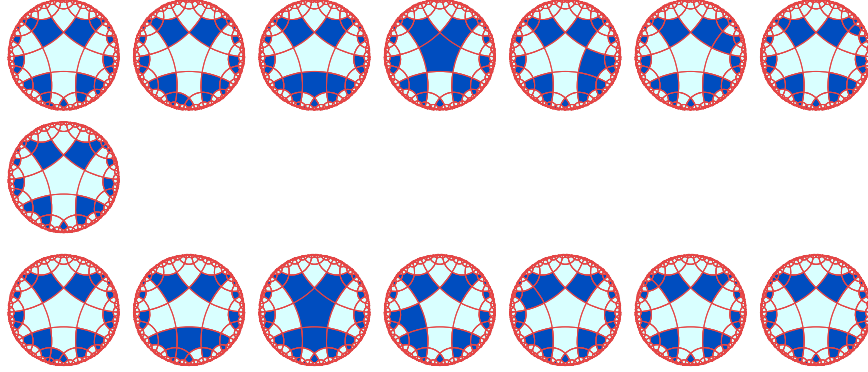


Figure 21 The selector of the round about. Above: a single locomotive enters the selector. It leaves it to go on its way on the right track. Below: a double locomotive enters the selector. It leaves it on the way going to the next selector.

Table 26 Rules for the selector: the case of a simple locomotive.

230	<u>WWWBWBWBWW</u>	239	<u>WBWWWWBBBW</u>	248	<u>WBWBWBWBWB</u>	257	<u>WBWBWBWBWB</u>
231	<u>BWBWBWBWBWB</u>	240	<u>WBWBWBWBWB</u>	249	<u>WBWBWBWBWB</u>	258	<u>BWBWBWBWBWB</u>
232	<u>WBWBWBWBWB</u>	241	<u>BWBWBWBWBWB</u>	250	<u>BWBWBWBWBWB</u>	259	<u>BWBWBWBWBWB</u>
233	<u>WWWBWBWBWB</u>	242	<u>WBWBWBWBWB</u>	251	<u>WBWBWBWBWB</u>	260	<u>WWWBWBWBWB</u>
234	<u>WBWBWBWBWB</u>	243	<u>WBWBWBWBWB</u>	252	<u>BWBWBWBWBWB</u>	261	<u>WBWBWBWBWB</u>
235	<u>BWBWBWBWBWB</u>	244	<u>BWBWBWBWBWB</u>	253	<u>BWBWBWBWBWB</u>	262	<u>BWBWBWBWBWB</u>
236	<u>WBWBWBWBWB</u>	245	<u>WWWBWBWBWB</u>	254	<u>WWWBWBWBWB</u>	263	<u>BBWBWBWBWB</u>
237	<u>BWBWBWBWBWB</u>	246	<u>BWBWBWBWBWB</u>	255	<u>BWBWBWBWBWB</u>	264	<u>BWBWBWBWBWB</u>
238	<u>BWBWBWBWBWB</u>	247	<u>BWBWBWBWBWB</u>	256	<u>BWBWBWBWBWB</u>		

Table 27 Rules for the selector: the case of double locomotive.

265	<u>BWBWBWBWBWB</u>	269	<u>WBWBWBWBWB</u>	273	<u>BWBWBWBWBWB</u>	276	<u>BBWBWBWBWB</u>
266	<u>BWBWBWBWBWB</u>	270	<u>BWBWBWBWBWB</u>	274	<u>BBWBWBWBWB</u>	277	<u>WBWBWBWBWB</u>
267	<u>BBWBWBWBWB</u>	271	<u>WBWBWBWBWB</u>	275	<u>WWWBWBWBWB</u>	278	<u>BWBWBWBWBWB</u>
268	<u>BWBWBWBWBWB</u>	272	<u>BWBWBWBWBWB</u>				

Note the very symmetric structure of the exit configurations when no locomotive is around. As shown by the figures, the cells of tracks involved in this structure are 8(3), 3(3), 1(3) and 0(0) for the arrival and then 1(2), 2(2), 5(2) and 13(2) for the path leading to the next selector and 1(4), 2(5), 7(5) and 20(5) for the track leading to the continuation of the required path. The milestones are 20(3), 23(3), 7(3), 10(3), 2(3) and 2(4) for the entry path, 1(1), 6(2), 4(1), 14(2), 12(1) and 35(2) for the path leading to the next selector and 1(5), 6(5), 3(5), 19(5), 8(5) and 53(5) for the track to the required path. We remain with the milestones of cells 1(2) and 1(4) as these cells play the selection

role: besides the already mentioned 1(1) and 2(3), we have 5(3), 3(2) and 10(2) for 1(2); besides the already mentioned 1(5) and 2(4), we have 5(4), 7(4), 18(4), 4(4) and 10(4) for 1(4).

Table 28 *Execution of the rules of the selector when it is crossed by a simple locomotive.*

	0	1	2	3	4	5
1(3)	234	240	244	249	191	234
0(0)	230	230	242	247	254	260
1(2)	232	232	243	248	232	232
2(2)	59	59	59	25	59	59
1(4)	236	236	245	251	255	261
2(5)	66	66	66	35	72	80
2(3)	235	241	131	250	235	235
2(4)	237	237	246	252	256	237

The conservative rule for 0(0) is rule 230, wwwbbwbbww, and we can see that the milestones of the cell are neighbours 4, 5, 7 and 8. Rule 232, wwbwbwbbww, and rule 236, wwbwbwbbww, are the conservative rules for 1(2) and 1(4) respectively. Note that they are rotated forms of one another. When the locomotive is a simple one, 1(2) remains white while the locomotive passes by. This can be checked on rule 243, wwbwbwbbww, and on rule 248, wwbwbwbbww, the locomotive being seen in the neighbour 10 of 1(2) for the first rule and in its neighbour 5 for the second one. For the same passage, 1(4) very differently reacts: rule 245, wwbwbwbbww, can see the locomotive through its vertex-neighbour 8 while rule 251, wwbwbwbbww, now seeing the locomotive in the neighbour 2 of the cell makes it become black. This can be checked on Table 28.

Table 29 *Execution of the rules of the selector when it is crossed by a double locomotive.*

	0	1	2	3	4	5
1(3)	240	265	267	228	234	234
0(0)	230	242	266	271	275	230
1(2)	232	243	193	273	277	232
2(2)	59	59	25	135	61	71
1(4)	236	245	269	236	236	236
2(5)	66	66	35	66	66	66
2(3)	241	174	268	274	235	235
2(4)	237	246	270	237	237	237

When the locomotive is double, 1(4) remains white. It is witnessed by rule 245 again, the locomotive is in the neighbour 8 of the cell, and then by rule 269, wwbwbwbbww, applied when the locomotive is in the neighbours 2 and 8 of the cell. But this time, 1(2) behaves differently: rule 243 again applies as the cell can only see the front of the locomotive through a vertex neighbour:

it cannot know whether it is simple or double. At the next time, rule 193, wwbwbwbwbwb, applies as the locomotive can be seen in the neighbours 5 and 10 respectively, making the cell becoming black.

3.4 Flip-flop and active memory switch

In Section 2 we have seen that we can implement both a flip-flop and the active memory switch by using the same basic devices provided that they are suitably arranged. Figure 22 shows how the controller is working, depending on its cell 1(1) whose state we call the **colour** of the controller. Figure 23 shows how the colour of controller is changed by the arrival of the appropriate signal: we remind the reader that the signal has the form of a simple locomotive. Table 30 gives the rules corresponding to the illustration given by Figures 22 and 23. Also, Table 31 gives the traces of executions of the rules applied to significant cells when the locomotive arrives to the controller while Table 32 gives similar traces when the signal arrives to the controller in order to change its colour.

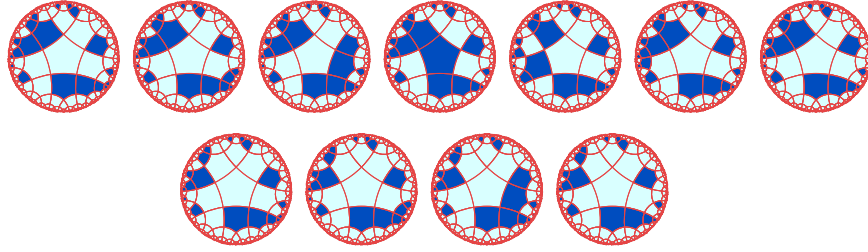


Figure 22 *The controller of the flip-flop and of the active memory switch. Above, the controller is black: the locomotive passes without problem. Below, the controller is white: it stops the locomotive which vanishes.*

Note that the tracks which crosses the controller passes through the cells 8(4), 3(4), 1(4), 0(0), 1(2), 3(2) and 8(2) in this order. The signal for changing the colour of the cell 1(1) passes through the cells 16(1), 6(1) and 2(1). The milestones of the path which crosses the controller are 54(4), 57(4), 20(4), 23(4), 7(4), 10(4), 2(4), 2(5), 1(3), 1(1), 4(2), 2(2), 9(2), 7(2), 20(2) and 22(2). The milestones for the path followed by the signal are the cells 41(1), 44(1), 15(1), 18(1) and 2(5). There are three additional milestones which are specific to the controller: the cells 4(1), 10(1) and 6(2).

Figure 22 shows us that when the cell 1(1) is black, the locomotive passes through the tracks, because 1(1) constitutes a milestone of the track. When 1(1) is white, the expected milestone being missing, the locomotive is stopped, see rule 315, wwbwbwbwbwb. The rule shows that the cell 0(0) remains white despite the fact that the locomotive is seen in the neighbour 3 of the cell. Indeed, the conservative rule for 0(0) when the cell 1(1) is white is rule 279, wwbwbwbwbwb. That rule can be compared with the one which is used when the cell 1(1) is black, namely rule 58, wwbwbwbwbwb.

Table 31 gives the traces of execution for the rules which apply to the cells 1(4), 0(0) and 1(2) for the tracks, the cells 2(2) and 1(1) for milestones and the cell 2(1) for a witness of transformations induced by the passage of the locomotive.

Table 30 *Rules for the controllers of the flip-flop and the active memory switch.*
managing the control

removing lock			setting lock		
279	<u>W</u> <u>B</u> <u>W</u> <u>W</u> <u>B</u> <u>W</u> <u>B</u> <u>B</u> <u>W</u>	286	<u>W</u> <u>B</u> <u>W</u> <u>W</u> <u>W</u> <u>W</u> <u>B</u> <u>W</u>	293	<u>W</u> <u>B</u> <u>W</u> <u>W</u> <u>B</u> <u>W</u> <u>W</u> <u>B</u> <u>W</u>
280	<u>W</u> <u>W</u> <u>W</u> <u>B</u> <u>B</u> <u>W</u> <u>W</u> <u>B</u> <u>W</u>	287	<u>W</u> <u>W</u> <u>W</u> <u>B</u> <u>B</u> <u>W</u> <u>W</u> <u>B</u> <u>B</u>	294	<u>B</u> <u>W</u> <u>W</u> <u>B</u> <u>B</u> <u>W</u> <u>W</u> <u>B</u> <u>W</u>
281	<u>W</u> <u>W</u> <u>W</u> <u>B</u> <u>W</u> <u>W</u> <u>W</u> <u>B</u> <u>B</u> <u>W</u>	288	<u>W</u> <u>W</u> <u>W</u> <u>B</u> <u>W</u> <u>W</u> <u>W</u> <u>B</u> <u>W</u>	295	<u>W</u> <u>B</u> <u>W</u> <u>B</u> <u>W</u> <u>W</u> <u>W</u> <u>B</u> <u>B</u> <u>W</u>
282	<u>W</u> <u>W</u> <u>W</u> <u>W</u> <u>B</u> <u>B</u> <u>W</u> <u>W</u> <u>W</u>	289	<u>W</u> <u>B</u> <u>W</u> <u>W</u> <u>W</u> <u>B</u> <u>B</u> <u>B</u> <u>W</u>	296	<u>B</u> <u>B</u> <u>B</u> <u>W</u> <u>W</u> <u>B</u> <u>W</u> <u>W</u> <u>W</u> <u>B</u>
283	<u>W</u> <u>W</u> <u>B</u> <u>W</u> <u>W</u> <u>W</u> <u>W</u> <u>W</u>	290	<u>W</u> <u>B</u> <u>W</u> <u>B</u> <u>B</u> <u>W</u> <u>W</u> <u>B</u> <u>W</u>	297	<u>W</u> <u>W</u> <u>B</u> <u>W</u> <u>W</u> <u>B</u> <u>W</u> <u>W</u> <u>W</u>
284	<u>B</u> <u>W</u> <u>B</u> <u>W</u> <u>W</u> <u>W</u> <u>B</u> <u>B</u> <u>B</u>	291	<u>B</u> <u>W</u> <u>B</u> <u>W</u> <u>W</u> <u>W</u> <u>B</u> <u>B</u> <u>W</u>	298	<u>B</u> <u>W</u> <u>B</u> <u>W</u> <u>W</u> <u>B</u> <u>W</u> <u>B</u> <u>B</u> <u>B</u>
285	<u>W</u> <u>B</u> <u>B</u> <u>W</u> <u>B</u> <u>W</u> <u>W</u> <u>W</u> <u>W</u>	292	<u>W</u> <u>B</u> <u>B</u> <u>W</u> <u>W</u> <u>W</u> <u>W</u> <u>W</u>	299	<u>W</u> <u>B</u> <u>W</u> <u>W</u> <u>B</u> <u>W</u> <u>W</u> <u>B</u> <u>W</u>
managing the motion of the locomotive					
selected track			non-selected track		
305	<u>W</u> <u>B</u> <u>B</u> <u>W</u> <u>W</u> <u>B</u> <u>B</u> <u>W</u> <u>B</u> <u>W</u>	309	<u>W</u> <u>B</u> <u>B</u> <u>W</u> <u>B</u> <u>W</u> <u>B</u> <u>B</u> <u>W</u>	312	<u>W</u> <u>B</u> <u>B</u> <u>W</u> <u>B</u> <u>B</u> <u>W</u> <u>B</u> <u>W</u>
306	<u>B</u> <u>B</u> <u>W</u> <u>B</u> <u>B</u> <u>W</u> <u>W</u> <u>B</u> <u>W</u>	310	<u>B</u> <u>W</u> <u>W</u> <u>B</u> <u>B</u> <u>W</u> <u>W</u> <u>B</u> <u>W</u>	313	<u>B</u> <u>W</u> <u>B</u> <u>W</u> <u>B</u> <u>B</u> <u>B</u> <u>W</u> <u>B</u>
307	<u>W</u> <u>B</u> <u>B</u> <u>W</u> <u>W</u> <u>W</u> <u>B</u> <u>B</u> <u>B</u>	311	<u>B</u> <u>W</u> <u>B</u> <u>W</u> <u>W</u> <u>B</u> <u>W</u> <u>B</u> <u>B</u>	314	<u>B</u> <u>B</u> <u>B</u> <u>W</u> <u>B</u> <u>W</u> <u>B</u> <u>W</u> <u>B</u>
308	<u>W</u> <u>B</u> <u>B</u> <u>W</u> <u>W</u> <u>B</u> <u>W</u> <u>W</u> <u>B</u> <u>W</u>				

Table 31 *Execution of the rules of the controller when a simple locomotive crosses the structure.*

2(1) is black:						2(1) is white:		
	0	1	2	3	4	0	1	2
1(2)	25	25	25	26	311	59	59	59
0(0)	58	58	78	83	309	279	279	315
1(4)	35	41	42	48	35	35	41	42
2(2)	143	143	143	266	312	127	127	127
1(1)	294	294	294	306	310	280	280	280
2(1)	295	295	295	307	295	281	281	281

When the colour of the controller is black, the conservative rule for 1(1) is rule 294, BWWBBWWBW. When the locomotive crosses the controller, cell 1(1) can see it first through in its neighbour 1, rule 306, BBWBBWWBW, and then in its vertex-neighbour 6, see rule 310, BWWBBWWBW. That rule and Figure 22 show us that the cell 2(2) became white after seeing the locomotive in 0(0), which is the neighbour 7 of 2(2). Indeed, in this situation, rule 266, BWBWBBWWW, applies to 2(2). When 2(2) is white, applying rule 312, WBBWBBWBW, allows us to restore the black state in the cell. This problem comes from the fact that rule 266 was needed in the selector for the cell 0(0) when a double locomotive crosses that cell. Note that after rule 310, rule 294 again applies to the cell 1(1).

Figure 23 illustrates the action of the signal on the cell 1(1): when the cell is black, the signal turns it to white and conversely. Table 32 displays which rules are used to which cells when the signal arrives in order to change the colour of the cell 1(1). The conservative rules for the cell 1(1) are rule 294, we have

already seen, when the cell is black, it is rule 280, $\underline{wwwbbwwwbw}$, when the cell is white. We can see that the neighbourhood is the same for rules 294 and 280: the milestones are in the neighbours 4, 5 and 9 of the cell, and the difference lies in the state of the cell itself. The change from black to blank is performed by rule 302, $\underline{wbwbbwwwbw}$, which can see the signal in its neighbour 2. The reverse change is obtained by rule 290, $\underline{wbwbbwwwbw}$, again as the signal is seen in the neighbour 2 of the cell.

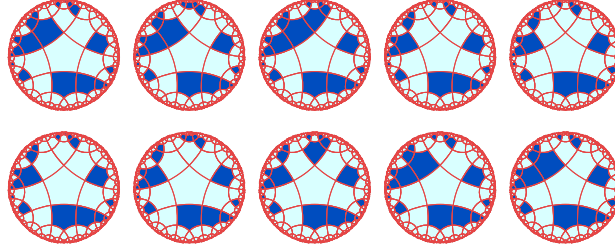


Figure 23 The controller of the flip-flop and of the active memory switch. The signal arrives to change the selection. Above, the controller is changed to white. Below, it is changed to black.

Table 32 Execution of the rules of the controller when a signal comes for changing the colour of the controller.

	from black to blank:					from blank to black:			
	0	1	2	3		0	1	2	3
6(1)	41	42	48	35		41	42	48	35
2(1)	295	300	303	281		281	287	291	295
0(0)	58	58	191	279		279	279	289	58
2(2)	143	143	143	127		127	127	127	143
1(1)	294	294	302	280		280	280	290	294
1(5)	299	299	190	286		286	286	293	299

Other neighbours of the cell 1(1) can see those changes. As an example which is not given in Table 31, we take the cell 1(5) which is always blank. The conservative rule is rule 299 $\underline{wbwwwbwwwbw}$, when the cell 1(1) is black, and it is rule 286 $\underline{wbwwwbwwwbw}$, when 1(1) is white. When the locomotive crosses the controller in the black colour, rule 305, $\underline{wbwwwbwwwbw}$, can see the locomotive in its neighbour 7, then rule 308, $\underline{wbwwwbwwwbw}$, can see it in its neighbour 1 and then, rule 299 is again applied. When 1(1) is white, rule 316, $\underline{wbwwwbwwwbw}$, is applied when the locomotive appears: it can be seen in its neighbour 7 and then the locomotive disappeared, so that rule 286 again applies. Table 32 shows the rules applied when the signal arrives, changing the colour of the cell 1(1).

3.5 The passive memory switch

As we said in Section 2, the controller of the passive memory switch works in a different way than that of the active switches. For this reason, we call it the **controller-sensor** as it does not stop the locomotive when it runs through the non-selected track. As in the case of the active switches, this controller is characterized by the state of a cell, again the cell 1(1), see Figure 24. Here, the cell 1(1) is called the **sensor** as its role is to detect a passage, not to stop it. As in the case of the controller, we say that the state of 1(1) is the **colour** of the sensor. The tracks goes along the cells 33(2), 12(2), 4(2), 1(2), 0(0) and 1(4), 4(4), 12(4) and 33(4) in this order. This order is the opposite of what he have seen for the passage of a locomotive through the controller of an active switch. This raises no problem thanks to the possibility to use round-about if needed: the round-about do not use the controllers nor the controller-sensors. The difference of direction induces a sharp difference in the milestones. Here the milestones of the track of the locomotive are: 34(3), 86(2), 13(3), 31(2), 5(3), 10(2), 2(3), 2(2), 2(5), 3(4), 5(5), 11(4), 13(5), 32(4), 34(5) and 87(4). Now, there is a common point between the controller-sensor and the controller of the active switches: it is the path of the signal for changing the colour of the sensor. The path and the milestones are the same. However a difference again: here, in the case of the controller-sensor, the colour is changed only if the sensor is black. When it is white, it is changed to black by the very passage of the locomotive. Here, there are also additional milestones: they consists of the cells 7(2), 6(2), 4(1), 10(1) and 2(4). Those actions are illustrated by Figure 24. Table 33 gives all the new rules induced by those constraints.

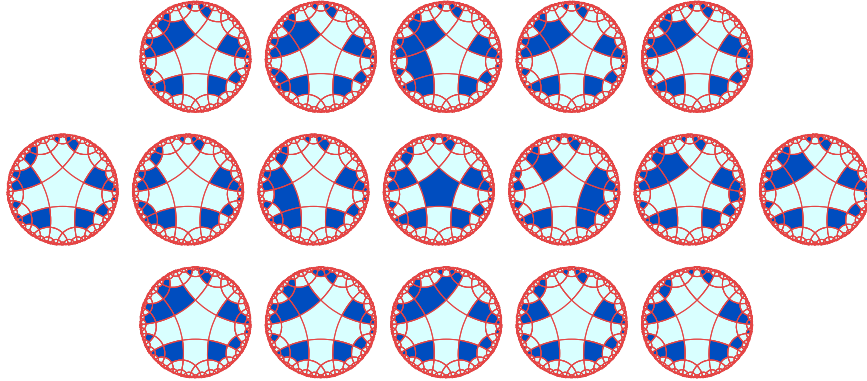


Figure 24 *The controller-sensor of the passive memory switch. Above: a single locomotive passively crossed the switch through its selected track. Middle row: the locomotive crossed the passive switch through the non-selected track. Below: the change of signal induced by a passive crossing through the non-selected track.*

Consider the case when the sensor is white. The conservative rule for 0(0) is

rule 317, wwwwwwBWB. Its milestones are in its vertex-neighbours 6, 8, 9 and 10.

Table 33 *Rules for the passive memory switch.*

rules for the control-sensor			
non-selected track		selected track	
317	<u>wwwwwwBWB</u>	330	<u>wwwwwwBWB</u>
318	<u>wwwwwwBWB</u>	331	<u>wwwwwwBWB</u>
319	<u>wwwwwwBWB</u>	332	<u>wwwwwwBWB</u>
320	<u>wwwwwwBWB</u>	333	<u>wwwwwwBWB</u>
321	<u>wwwwwwBWB</u>	334	<u>wwwwwwBWB</u>
322	<u>wwwwwwBWB</u>	335	<u>wwwwwwBWB</u>
323	<u>wwwwwwBWB</u>	336	<u>wwwwwwBWB</u>
324	<u>wwwwwwBWB</u>	337	<u>wwwwwwBWB</u>
325	<u>wwwwwwBWB</u>	338	<u>wwwwwwBWB</u>
326	<u>wwwwwwBWB</u>	339	<u>wwwwwwBWB</u>
327	<u>wwwwwwBWB</u>	340	<u>wwwwwwBWB</u>
328	<u>wwwwwwBWB</u>	341	<u>wwwwwwBWB</u>
329	<u>wwwwwwBWB</u>	342	<u>wwwwwwBWB</u>
		343	<u>wwwwwwBWB</u>
		344	<u>wwwwwwBWB</u>
		345	<u>wwwwwwBWB</u>
		346	<u>wwwwwwBWB</u>
		347	<u>wwwwwwBWB</u>
		change of selection	
		348	<u>wwwwwwBWB</u>
		349	<u>wwwwwwBWB</u>
		350	<u>wwwwwwBWB</u>
		351	<u>wwwwwwBWB</u>
		352	<u>wwwwwwBWB</u>

Table 34 displays the rules used for the cells of the controller-sensor when the locomotive passes through the non-selected track. When the locomotive arrives at the cell 0(0), it is seen from the neighbour 3 of the cell see rule 325, wwwwwwBWB : here, the side 1 of cell 0(0) is shared with 1(5). Then, rule 328, wwwwwwBWB, makes the locomotive leave the cell. Rule 25, wwwwwwBWB, witnesses that the locomotive left 0(0) as it can be seen in the neighbour 5 of 0(0). The rule also witnesses that the sensor turned to black: the cell 1(1) is the neighbour 2 of 0(0) which is now black. Rule 25 is a rule of the vertical track is also applied here because at that very moment, the cell 2(2) became white and the cell 2(2) is the vertex-number 8 of the cell which, usually, is black. After that, rule 289, wwwwwwBWB, is applied. It is the conservative rule of 0(0) when the sensor is black: remember that the cell 1(1) is the neighbour 2 of 0(0). The rule also witnesses that the cell 2(2) returned to black.

Table 34 *Execution of the rules for the controller of the passive memory switch when the sensor is white.*

	0	1	2	3	4	5
4(2)	26	27	28	25	25	25
2(2)	234	322	244	249	336	228
0(0)	317	317	325	328	25	289
4(1)	321	321	321	332	339	341
4(4)	35	35	35	35	41	42
2(2)	127	127	327	247	337	143
1(1)	280	280	326	329	334	294

We leave to the reader the examination of the application of the rules given by Table 33 to the cells indicated in Table 34. The first two rows of Figure 24 are of help for such a task.

Table 35 gives traces of execution for the rules applied to the cells when the locomotive crosses a black sensor and for those concerned by the change of the

sensor from black to white. In the latter case, the comparison of the right-hand side part of Table 35 with the left-hand side part of Table 32 show us that the same rules apply. This explains why the last row of Figure 24 is very similar to that of the first line of Figure 23 for the part of these figures involved by the cells concerned by the signal: the track where the signal goes, their milestones and the cell 1(1) with its neighbours. Similarly, the behaviour of the cell 2(2) is the same in the first line of Figure 22 and in the first one of Figure 24.

An important difference should also be noticed between the first line of Figure 24 and the first line of Figure 22. It is the fact that the sensor is white in the first case while it is black in the second one. Also, in the first case the sensor changes its colour while its colour is unchanged in the second one. This explains why a few rules in the Tables 34 and 35 are taken from Table 33.

Table 35 *Executions of the rules for the control of the passive memory switch when the sensor is blue.*

the locomotive passes					change from blue to red				
	0	1	2	3		0	1	2	3
4(2)	26	27	28	25	6(1)	41	42	48	35
1(2)	228	343	346	228	2(1)	295	300	303	281
0(0)	289	289	344	289	2(2)	143	143	143	127
2(2)	143	143	347	143	1(1)	294	294	302	280
1(1)	294	294	345	294					

As examples, consider the cells 0(0), 1(1) and 2(2) when the locomotive passes while the sensor is black. As already mentioned, the conservative rule for 0(0) when the cell 1(1) is black is rule 289. When the locomotive is seen by 0(0), the locomotive is in the neighbour 3 of 0(0), so that rule 344, wwbbwwbwwbww, is applied. At the next time, as the locomotive vanished, rule 289 is again applied. In the case of 1(1) when it is black, the conservative rule is rule 294, bwwbbwwbwwbww, a rule from Table 30. When the cell can see the locomotive which appears as the neighbour 6 of 1(1), rule 345, ewwwbbwwbwwbww, is applied. Again, as the locomotive vanishes at the next time, rule 294 is again applied. At last, the conservative rule for 2(2) is rule 143, ewbwwbbwwbwwbww, a rule for the horizontal tracks. Here too, the presence of the locomotive is noticed for one time only: the locomotive is in the neighbour 1 of 2(2): rule 347, bbwwbbwwbwwbww, is applied. At the next time, rule 143 is applied: the locomotive vanished.

The tables of this section completes the proof of Theorem 1. □

3.6 About rotation invariance

We already mentioned in Section 3 that the automaton we constructed to prove Theorem 1 is not rotation invariant. We used this property in order to suitably number the sides of each cell. Among the rules displayed in the various tables of Section 3, many of them are rotated forms of other rules. As an example, rules 3,

39, 126 and 286, *i.e.* WBWWWWWWWW, WBWWWWWWWW, WWWWBWWWWW and WWBWWWWWWW, respectively, are rotated form of each other. They apply to white cells with a single black neighbour, neighbour 1, 2, 5 and 3 respectively. We shall say that in the corresponding neighbourhoods there is a shift by 1, 4 and 2 for rules 39, 126 and 286 respectively with respect to that of rule 3.

Table 36 *Rules which contradict rotation invariance, part I.*

21	<u>WBWBWBWBWB</u>	21(3)	Vdd	65	<u>WBWBWBWBWB</u>	*1	Has	3
		7(4)	Hwcd			*2	Had	
		21(3)	fxld			*3	Hwas	
		21(3)	fxrd			*4	Hwad	
		21(3)	D					
		8(2)	Cb					
30	<u>WBWWBWBWBW</u>	*5	Vdd	287	<u>WBWBWBWBWB</u>	2(1)	Cws	3
		*6	Hcd					
		*7	Had					
		*8	fxld					
		*9	fxrd					
		*9	D					
		*10	Sls					
		*10	Sld					
		*11	Cbs					
		*11	snbs					
51	<u>WBWBWBWBWB</u>	33(1)	Vud	124	<u>WBWBWBWBWB</u>	*12	Hcs	2
		5(4)	Had			*13	Hcd	
						*14	Hwcs	
						*15	Hwcd	
				300	<u>WBWBWBWBWB</u>	2(1)	Cbs	1
						2(1)	Snbs	
80	<u>BWBWBWBWBW</u>	*16	Has	252	<u>BWBWBWBWBW</u>	2(4)	Sls	3
		*17	Hcs					
		*18	Hwas					
		*19	Hwcs					
		2(5)	Sls					
		2(5)	fk					
112	<u>BWBWBWBWBW</u>	*20	Had	339	<u>BWBWBWBWBW</u>	1(4)	Snw	2
		1(1)	Hcd					
*1	18(1), 10(1), 5(2), 7(3), 10(3)			*2	10(1), 5(2), 7(3), 10(3)			
*3	10(4), 5(5), 2(1), 7(2), 10(2)			*4	5(5), 2(1), 7(2), 10(2)			
*5	11(1), 3(1), 1(5), 2(4), 10(3), 23(3)							
*6	9(3), 6(3), 4(2), 7(2), 12(1), 9(1), 20(1), 17(1)							
*7	9(2), 6(2), 4(1), 4(5), 7(5), 12(4), 9(4)							
*8	11(1), 3(1), 2(4), 10(3), 23(3)			*9	2(4), 10(3), 23(3)			*10 6(4) *11 3(1)
*12	10(3), 7(3), 5(2), 10(1), 18(1)			*13	7(3), 5(2), 10(1), 18(1), 15(1)			
*14	10(2), 7(2), 2(1), 5(5), 10(4)			*15	7(2), 2(1), 5(5), 10(4)			*16 7(1), 2(2)
*17	3(3), 4(1)			*18	2(5), 1(1)			*19 3(2), 4(4) *20 1(1), 1(3)

Now, there are 14 pairs of rules in which the neighbourhoods are rotated, the state of the cell is the same but its new state is different. Tables 36 and 37 give the rules together with the cells to which they apply and in which part of the circuit. The part of the circuit is identified by a few letters as follows. For the tracks, **V** means vertical and **H** means horizontal; **d** as second letter means down while **u** means up; **d**, **s** as a last letter indicates a double, simple locomotive, respectively; **c**, **a** means a clockwise, counter-clockwise running, respectively; **w** means that the horizontal goes around a white node; when **w** is missing, this means that the horizontal goes around a black node. The fixed switch is represented by **fx**, **l**, **r** indicates from the left-, right-hand side respectively. The fork is represented by **fk**, the doubler by **D**. The controller is denoted by **C** while the controller-sensor is defined by **Sn**, **b**, **w** indicate the colour of the controller or the sensor and **s** indicates the run involving the signal. At last, **S1** denotes the selector, **s** and **d** indicating whether it is crossed by a simple or a double locomotive.

Table 37 *Rules which contradict rotation invariance, part II.*

147	<u>BWB</u> BBBWWB	1(3)	Hcd	339	<u>BBW</u> BWBWWWW	1(4)	Snw	1
202	<u>WWB</u> BWBWWB	0(0)	D	242	<u>WWB</u> BWBWWB	0(0)	Sls	2
						0(0)	seld	
209	<u>BBW</u> BWBWWWW	1(1)	D	265	<u>BWB</u> BWBWWBBB	1(3)	Sld	2
212	<u>WBW</u> BWBWWWW	1(1)	D	240	<u>WWB</u> BWBWWBBB	1(3)	Sls	2
						1(3)	seld	
226	<u>BWB</u> BWBWWWW	1(4)	fk	250	<u>BWB</u> BWBWWWWB	2(3)	Sls	4
246	<u>BWB</u> BWBWWWWB	2(4)	Sls	339	<u>BBW</u> BWBWWWW	1(4)	Snw	3
		2(4)	Sld					
251	<u>WWB</u> BWBWBWB	1(4)	Sls	277	<u>WBW</u> BWBWBWB	1(2)	Sld	4
256	<u>BWB</u> BWBWWWWB	2(4)	Sls	302	<u>BWB</u> BWBWWWW	1(1)	Cbs	4
						1(1)	Snbs	

The tables indicate that the relaxation of the rotation invariance plays an important role. However, the number of rules concerned is a bit less than ten percents of the rules. Also the high number of rotated forms of rules is an interesting feature. At last, besides the first rules numbered from 1 to 10, several rules are used in many different places. As an example, rule 190, WWBWWBWWB, is used in many places: 5(3) in horizontal tracks which clockwise go around a white node, both for the simple and the double locomotive. The rule is also applied to the cell 1(5) in the controller and in the controller-sensor when the black colour is changed to the white one. We have already seen that these motions of the signal are very similar in both those cases.

Three rules have all their rotated forms present in the tables: rule 16, rule 23 and rule 56. As an interesting example, we have chosen rule 16, BBWWBWWWWB, whose other forms are rule 81, BWBBBWWWWB, rule 93, BWBBBWWWWB, rule 136, BBWBBWWWWB and rule 320, BBBWBWWWWB. With respect to rule 16, the shifts applied to rule 81, 93, 136 and 320 are 3, 2, 4 and 1, respectively. Rule 16 is present very often, in many situations, the horizontal motions excepted. Rule 81 is ap-

plied to 1(1) and 1(3) each time once in the horizontal motions with the simple locomotive around a black node. Rule 93 is applied once to 1(1) in the same motions. Rule 136 is applied once to 1(3) in the clockwise motion only. Rule 320 is applied to 5(3) and 13(3), once in the sensor, in the passage of the locomotive, whatever the colour of the sensor.

It should be noted that the rules of Table 36 apply to many cells, in particular the first five pairs of the table. This is due to the fact that the rules mainly concern elements of the tracks. By contrast, in each pair of Table 37, each rule concerns at most two cells.

Conclusion

Several questions are raised by this result. How to reduce the number of states in this situation? Is it possible to obtain two states with a Moore neighbourhood and still observing rotation invariance? Still hard work ahead.

References

- [1] F. Herrmann, M. Margenstern, A universal cellular automaton in the hyperbolic plane, *Theoretical Computer Science*, **296**, (2003), 327-364.
- [2] M. Margenstern, *Cellular Automata in Hyperbolic Spaces, vol. 1, Theory*, Collection: *Advances in Unconventional Computing and Cellular Automata*, Editor: Andrew Adamatzky, Old City Publishing, Philadelphia, (2007), 422p.
- [3] M. Margenstern, *Cellular Automata in Hyperbolic Spaces, vol. 2, Implementation and computations*, Collection: *Advances in Unconventional Computing and Cellular Automata*, Editor: Andrew Adamatzky, Old City Publishing, Philadelphia, (2008), 360p.
- [4] M. Margenstern, A family of weakly universal cellular automata in the hyperbolic plane with two states, *arXiv:1202.1709*, (2012), 83p.
- [5] M. Margenstern, *Small Universal Cellular Automata in Hyperbolic Spaces: A Collection of Jewels*, Springer Verlag, (2013), 331p.
- [6] M. Margenstern, A Weakly Universal Cellular Automaton in the Pentagrid with Five States, *Lecture Notes in Computer Science*, C.S. Calude et al. (Eds.): Gruska's Festschrift, **8808**, (2014), 99-113.
- [7] M. Margenstern, A weakly universal cellular automaton in the heptagrid with three states, *arXiv:1410.1864v1*, (2014), 27p.
- [8] M. Margenstern, A weakly universal cellular automaton with 2 states in the tiling $\{11, 3\}$, *Journal of Cellular Automata*, to appear.

- [9] M. Margenstern, A weakly universal cellular automaton in the pentagrid with three states, *arXiv:1510.09129*, (2015), 40p.
- [10] M. Margenstern, Cellular Automata in Hyperbolic Spaces, chapter in *Advances in Unconventional Computing*, Springer, to appear.
- [11] M. Margenstern, Y. Song, A new universal cellular automaton on the pentagrid, *Parallel Processing Letters*, **19**(2), (2009), 227-246.
- [12] M.L. Minsky, *Computation: Finite and Infinite Machines*, Prentice-Hall, Englewood Cliffs, NJ, 1967.
- [13] I. Stewart, A Subway Named Turing, Mathematical Recreations in *Scientific American*, (1994), 90-92.